STUDY OF A RANDOM ACCESS PACKET NETWORK USING THE SPREAD-SPECTRUM TECHNIQUE

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

BISWAJIT NANDY

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
AUGUST, 1984

To

My Parents

CERTIFICATE

It is to certify that the thesis entitled 'STUDY OF A RANDOM ACCESS PACKET NETWORK USING THE SPREAD-SPECTRUM TECHNIQUE' submitted by Mr. Biswajit Nandy, Roll No. 8210407, for the partial fulfilment of the requirements for the award of M.Tech. degree, has been carried out under my supervision. According to my knowledge, it has not been submitted elsewhere for an award of a degree.

27/8/84

(J. Das)
Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur

ACKNOWLEDGEMENTS

I express my deep sense of gratitude and appreciation to Dr. J. Das and Dr. R. Raghuram for their invaluable guidance and constant encouragement throughout the course of my thesis work.

I am highly obliged to Dr. R.N. Biswas and Dr. P.K. Chatterjee for making available the laboratory facilities without which this project could not have been implemented.

I am grateful to my friend, Gautam Deb for his delightful company and constant cooperation during the period of my work.

I express my thanks to my friends, Mukund , Deshpande, Ravikumar, Gayaprasad, Lt. Dutt, Satya and Sumit for many enlightening 'bullsessions' which made last two years memorable.

Special thanks to Somnath and Mrityunjoy for their help in preparing the thesis.

Finally, I thank Mr. J.S. Rawat for his excellent typing of the manuscript.

Aug. 25,1984

(Biswajit Nandy)

ABSTRACT

A microprocessor based interface message processor (IMP) and spread spectrum transmitter, receiver have been implemented with a view to study the performance of a spread spectrum multiple access (SSMA) network.

The well known multi-point line, supports a data communication system where all the stations are in contact with each other over a single medium of communication. If more than one user is allowed to transmit at any time there will be collisions and colliding packets will be destroyed. To overcome this each user is assigned a code in the form of a maximal length pseudo-noise sequence (PNS) in which all messages addressed to it must be modulated. The maximal length PN sequences have very low cross-correlation. So within the limit of the error due to other users' interference, the packet is reliably transferred. To assure error-free communication, a HDLC-like data link control protocol has been implemented. The SSMA node is interfaced to a host system with ASCII keyboard and TTY in which an interactive users' interface has been developed.

The bit error probability as a function of number of users is measured. From this the average throughput and delay of the system for finite number of users is evaluated. It is found that the spread spectrum technique in random access environment improves the performance over other popular random access techniques. The maximum throughput obtained is 2.4 packets/unit time.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 THE RANDOM ACCESS COMMUNICATION PROBLEM:

Communication engineers have a long acquaintance with the 'multiple-access' problem, i.e. the problem of providing the means where by many senders' information can share a common communication resource. The classical solution is to do some form of multiplexing. When there are only a small and fixed number of users, each of which has a heavy load of traffic, the Frequency Division Multiplexing (FDM) is simple and efficient allocation mechanism. As the computer traffic is bursty in nature, the most of the bandwidth is wasted. The Time Division Multiplexing increases the average delay and it requires costly equipments for synchronization. For low and uneven traffic the demand multiplexing can improve the performance.

In early 1970s, Abramson [1-3] devised a new random access method. Although his work, called ALOHA, used ground based radio packet broadcasting system, the basic idea is applicable to any system in which uncoordinated users are competing for the use of a single shared channel. The idea is to permit any user to seize the entire communication

resource when it happens to have information to transmit. There will be collisions and by perfect feedback arrangement the sender can always find out whether or not his packet was destroyed. In 1972 Roberts [2],[4] proposed the slotted ALOHA system to double the capacity of ALOHA. After that, many modifications were suggested to schedule transmissions so as to avoid collisions. The rules imposed to the process have a fundamental influence on the performance measures like throughput, delay etc.

In 1974, Tobagi [2], [5] suggested an elegant technique. The idea is to inhibit the transmission by sensing the channel when the channel is busy. This is known as 'carrier Sense Multiple Access'. By modifying the sensing strategy (listen while talk), Metcalfe and Boggs of Xerox [17] gave the idea of Ethernet.

For all the described systems, high throughput and low delay are in conflict. Good performance of one of them can be achieved only at the expense of the other. It is found that improvement in both can be achieved by some sort of channel coding. In 1981 Roychaudhuri [10] analysed the performance of random access spread spectrum multiple access systems. It shows the improvement of both the throughput and delay characteristics. Keeping an eye to these performance

measures [10,11] a spread spectrum coding [6-8] technique is used to recover some of the collided packets [9] and thereby increase the throughput. In addition it can provide lower average delay since more packets can be transmitted in a given time. It is found that the performance improves over other popular random access methods. Anti-jamming property, inherent security and greater immunity to interference and interception are the added advantages of spread spectrum system.

The objective of the project is to implement a random access network using the spread spectrum technique and to analyse its performance.

1.2 MULTIPLE ACCESS CAPABILITY OF SPREAD SPECTRUM:

Each user is assigned a spectrum sequence which is near orthogonal to that of any other user. The data is spread by the sequence. Such spectrally spread signals are added and transmitted over the channel. The signal is received along with the additive uncorrelated interference due to other users and noise, as is shown in Fig. 1.1. This composite signal is correlated with each of the SS sequences. It removes the coding from the wanted signal, leaving only the narrowband message signal, but unwanted signals are still spread over a wide band, so as to cause minimum interference (Fig. 1.2). The larger the ratio of the bandwidth of the

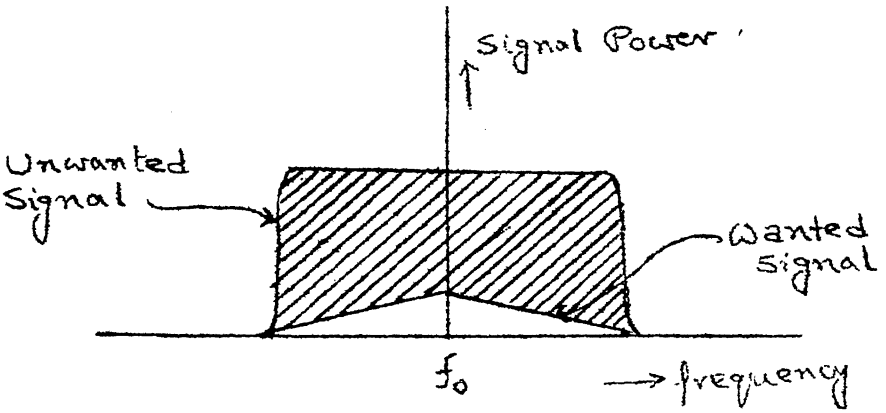


Fig 1.1 Received Signal with Interference and Noise

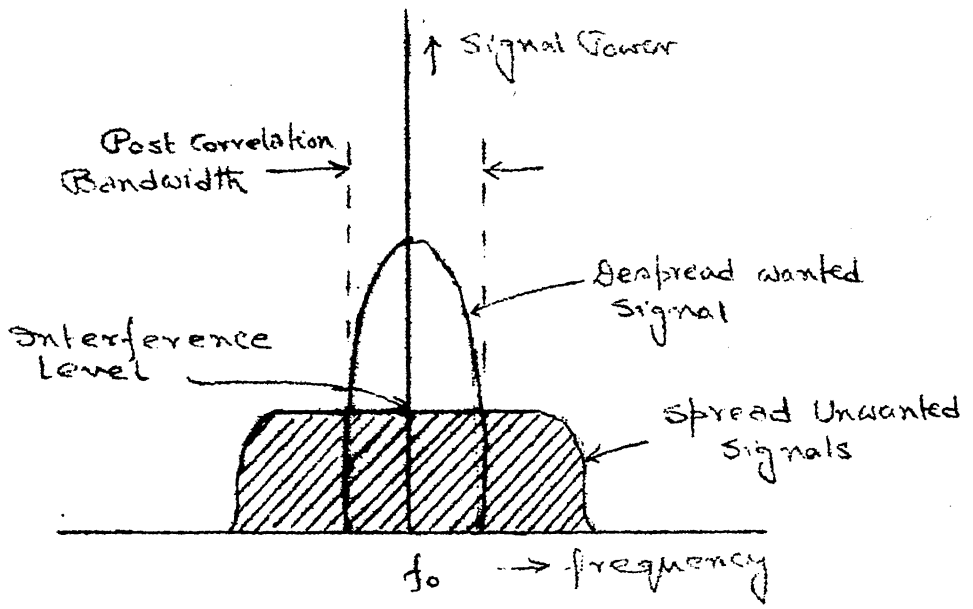


Fig 1.2 Correlator Output

spread signal to the information signal is the smaller the effect of unwanted signal interference. This ratio is defined as processing gain.

1.3 SYSTEM DESCRIPTION:

The assumed topology of the network is multipoint (Fig. 1.3). All the N stations are connected to same communicating medium. Any station can initiate transmission at any time and can communicate with any other station. Each station is assigned a unique code word from a set of N maximal length PN sequences to achieve spread spectrum. A station will thus be able to receive only transmissions that are spread using its assigned code, i.e. only those packets that are addressed to it. The bandwidth requirement is 127 times the actual signal bandwidth. When a multiple number of transmissions are present at a particular instant during reception of one of the packets by a receiver, the rest of the transmissions not addressed to it appear as noise and may introduce errors. The data-link control protocol takes care of these errors. So the host system is assured of error free transmission. The maximum allowable baud rate of the implemented system is 9.6 K baud.

The spread spectrum network can be efficiently implemented in local areas among large number of uncoordinated

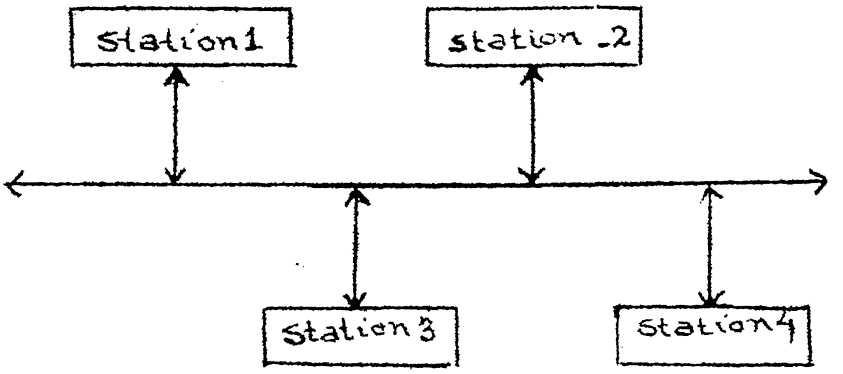


Fig 1.3a) Multipoint Network

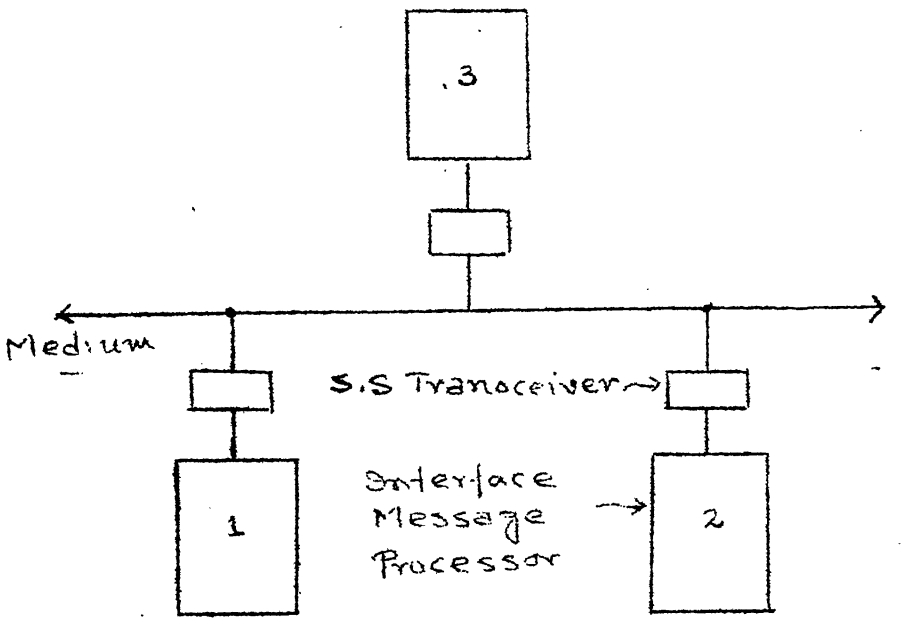


Fig 1.3 b) Jke Network Topology

users. It can be used for communicating among microcomputers, printing facilities, back-up storage devices and large control computers.

1.4 ASSUMPTIONS:

For analysis, let us make some reasonable and simplifying assumptions that will enable us to concentrate on the most significant aspect of the problem, i.e., the impact of SSMA on throughput and delay.

1. The processing time in the interface message processor is zero.
2. The propagation delays are very small compared to packet duration i.e. near-far effects are negligible.
3. Errors in the received packets are solely due to the simultaneous presence of many packets.
4. Probability of occurance of two or more packets addressed to the same station in a given time is negligible.
5. All the packets are of constant length.

1.5 THE THESIS OUTLINE:

Chapter 2 describes the functions performed by the implemented physical, data-link and network layer.

Chapter 3 gives the details of IMP implementation. It describes the hardware and the software required to support the functions of different layers.

Chapter 4 deals with the spread spectrum technique implemented to resolve the contention problem. It also describes the implementation details of the SS transmitter and receiver.

In Chapter 5 the performance of the system is evaluated. From the measured bit error probability the average throughput and the delay of the system is calculated. The theoretical values are also given.

In Chapter 6 the performance of the system is compared with the ALOHA and the CSMA techniques. Suggestions are given for further improvement.

CHAPTER 2

THE NETWORK ARCHITECTURE

This chapter describes the layered architecture of the network. It describes the functions performed by the implemented physical, data-link and network layers.

2.1 THE ISO REFERENCE MODEL:

The ISO reference model [16] is followed for the hierarchical organisation of the network. Here we have elaborate physical, data-link and network layers. The application presentation, session and transport layers are compressed into a simple users' interface. A more realistic users' interface can be added without affecting the rest of the system. The advantage of the layered architecture is that the design of any layer can be changed without affecting the other layers, provided proper interface between the layers is maintained. Fig. 1.1 shows the different layers.

2.2 THE PHYSICAL LAYER:

The links are serial. External clock is provided to all nodes. The transmission is baseband. A TTL high level signifies a '1' and a TTL low level a '0'. The data is encoded in Non-Return to Zero Inverted (NRZI). The maximum baud rate is 9.6K baud.

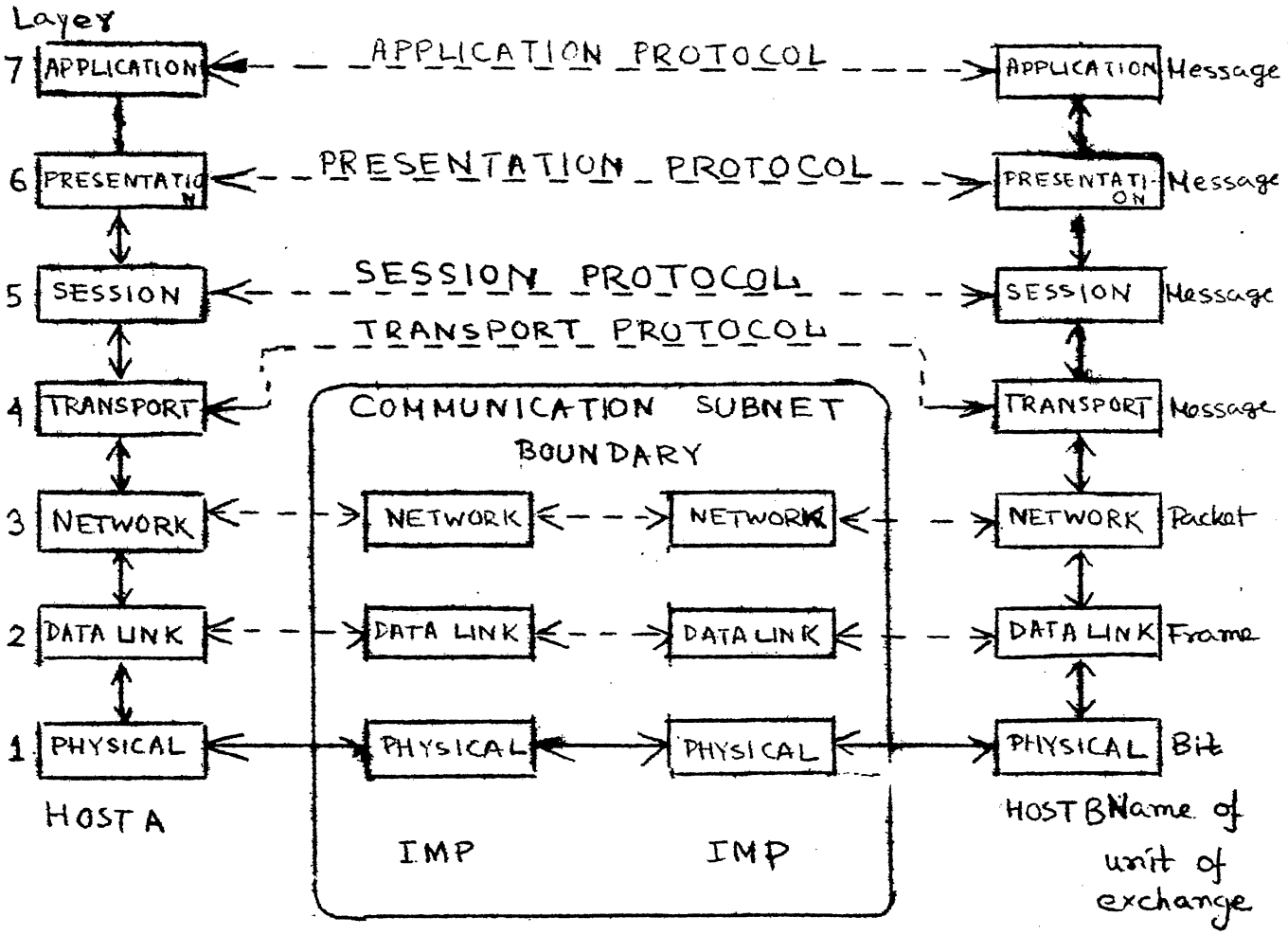


FIGURE: 2.1 : ISO-OSI REFERENCE MODEL

2.2.1 Facilities Assumed from Physical Layer:

The Data Link Control protocol assumes the following facilities: A facility of transmitting data with or without errors from one Interface Message Processor (IMP) to the other. It is sequential in nature in the sense that a bit which is transmitted before another cannot be received after the other (of course it is not claimed that all bits have to reach the destination !)

2.3 THE DATA LINK LAYER:

Data link protocols are among the oldest recognized communication protocols. It evolved continuously from the early primitive protocols through the widely implemented character-oriented protocols, to the increasingly popular bit-oriented protocols. The protocol provides a well defined set of rules which govern the interchange of information over interconnecting communication link between devices and users. The devices may be terminals, computers, message or packet switches, concentraters or any data terminal equipment. The user may be application software running in a processor or human operators interacting through a terminal device.

The basic procedure used in all of them is essentially the same, which can be stated as, 'keep transmitting the same frame again and again until you get an acknowledgement

for it. An acknowledgement implies correct receipt of the frame'.

The bit-oriented HDLC-like protocol has been implemented. The way the functions are accomplished varies with type and sophistication of the actual HDLC-protocol. The functions are described below.

2.3.1 Framing the Message:

All transmissions on data link are organized in a specific format. This format enables the receiving IMP to determine where the transmission starts and stops, whether the transmission is for that station, what actions are to be performed, specific information and the data used to check whether the frame was received without error.

The different fields of a frame is shown in Fig. 2.2.

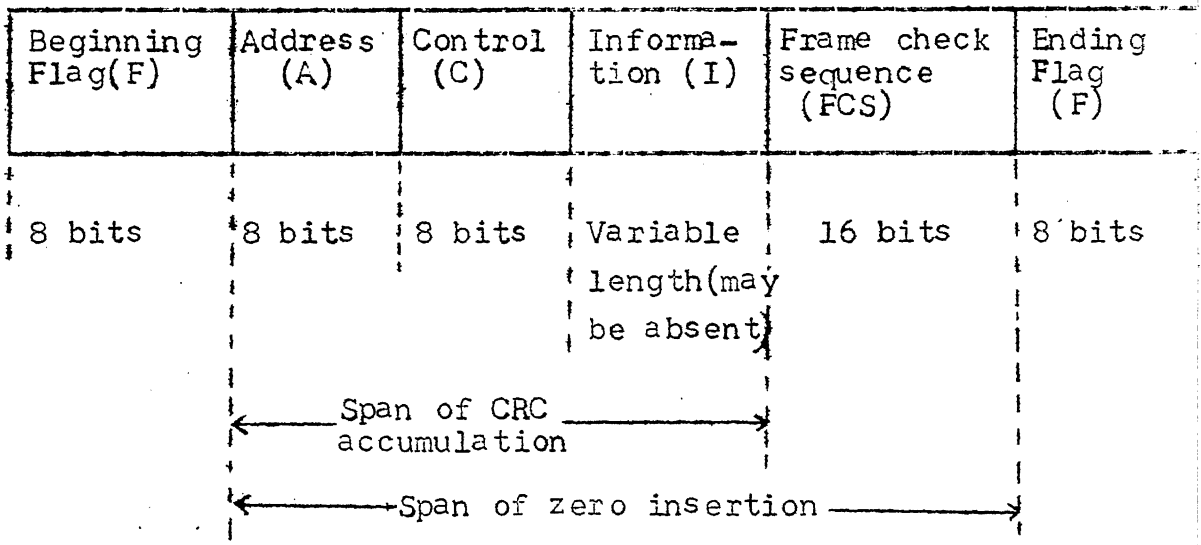


Fig. 2.2: Fields of the Transmitted Frame.

2.3.1.1 Flags:

The beginning flag and the ending flag enclose a frame. The beginning flag serves as a reference for the position of the address and control fields and initiates transmission error checking, the ending flag terminates the check for transmission errors. Both the flags have the binary configuration 0111 1110.

2.3.1.2 Address Field:

It follows immediately after the beginning flag (Fig. 2.1 The address field contains the address of the sender station. The receiver station needs this information for sending acknowledgement.

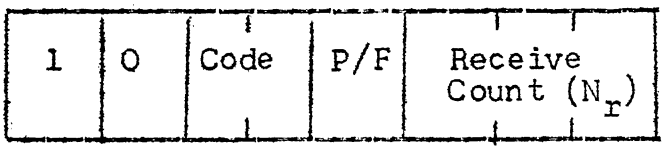
2.3.1.3 Control Field:

This field defines the function of the frame. The control field can be in one of the three formats: unnumbered, supervisory or information.

Unnumbered frames are used for initializing a station and controlling the response. Supervisory frames confirm received frames, convey ready or busy conditions. The control field of information frame contains send and receive counts (N_s and N_r). Sec. 2.3.3 describes the sequence control process using N_s and N_r counts.

Unnumbered

Supervisory



Information

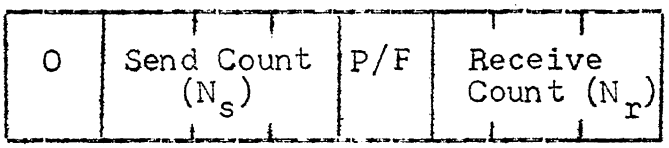


Fig. 2.3: Control fields in different formats

In the implemented version seperate acknowledgement is sent for the information frame. No piggy - backing facility is there. So the N_r bits of information frame are not used.

All the three control field formats contain a poll/final (P/F) bit. As all the stations can initiate transmission at any time, this bit is also of no use.

2.3.1.4 Information Field:

Following the control field, there may or may not be an information field. The information field does not have a set length. Long blocks of information are segmented into the packets of 256 bytes. Supervisory and unnumbered frames do not have this field.

2.3.1.5 Frame Check Sequence Field:

The purpose of this field is to check the received frame for errors that may have been introduced by the communication channel. The field contains a 16-bit check sequence that is the result of a computation on the contents of the A,C and I fields, at the transmitter. The computation used is cyclic redundancy checking (CRC).

The receiver performs a similar computation and checks its result. The receiver accepts no frame that is found to be in error. The FCS field is followed by the ending flag, closing the frame.

2.3.2 Transparency:

A frame is identifiable because it begins and ends with a flag and contains only nonflag bit patterns between the flags. Thus, no pattern of 01111110 (a flag) should be transmitted between the beginning and ending flags. This difficulty is overcomed by the zero-insertion process.

A binary '0' is inserted by the transmitter after any succession of five contguous 1's within the frame. Zero bit insertion is disabled when the flag is being transmitted. After testing for a '0' that follows five contiguous 1's. Inserted and removed '0's are not included in the frame check sequence computation. A '1' that follows five 1's is not removed.

2.3.3 Sequence Control:

A provision is made for transmitting a sequence of numbered information frames and making sure that they are received in the proper order. A station transmitting numbered information frames counts each such frames and sends the count with the frame. This count (N_s) is checked at the receiver for missing or duplicated frames.

A receiving station accepts each numbered information frame (error-free and in-sequence) and advances its receive count (N_r) for each such frame. If the received frame is error-free and in-sequence, a receiving station's N_r count is the same as the N_s count that it will receive in the next numbered information frame. The receiver confirms accepted numbered information frames by returning its N_r count to the transmitting station. The counting capacity for N_r and N_s

is eight. These counts are modulo in nature.

2.3.4 Abnormal Condition Recovery:

Almost all the link level abnormal conditions are recovered by retransmission at time out. Some examples are loss of response, out of sequence frame, CRC error, frame less than 32 bits long, acknowledgement not received.

2.3.5 Flow-control:

Receiver needs to be able to regulate the flow of information into the system in order to prevent being overwhelmed by incoming data. When the receive buffer is full it rejects the incoming data.

2.3.6 Initialization and Termination:

The initialization function deals with the establishment of an active Data Link connection over the physical path. The exchange of a supervisory frame and its acknowledgement establishes the readiness to receive and transmit. Following the transfer of the user's information, the link which was logically established by the initialization process, is terminated. Termination functions involve the tidying up the link by assuming that all data sent have been received. Then clearing the logical connection.

The above described functions taken together comprise the set of services that the Data Link Layer

makes available to the Network Layer. Now the network layer supports the intersystem communication.

2.4 NETWORK LAYER:

This layer determines the characteristic of the IMP-host interface. It accepts messages from the host, convert them to packets and arranges them in buffer. It transfers the packets to host in order.

The developed users' interface is very simple. The implementation details of the functions of different layers are described in Chapter 3.

CHAPTER 3

IMP IMPLEMENTATION DETAILS

This chapter describes the hardware and software implementation details. All the data and address bytes are mentioned in HEX. A brief description of the hardware is given. The logically important portions of the software is detailed by flow charts.

3.1 DESCRIPTION OF HARDWARE:

The conceptual block diagram of IMP and HOST is shown in Fig. 3.1. The interface message processor is based on the WORK STATION developed in the Deptt. of Electrical Engineering, I.I.T. Kanpur. The work station has got three cards. Of these, the processor card with INTEL 8085 CPU is used as the processor of the IMP. The lines from the CPU are brought out to a connector of which the following lines have been used; demultiplexed address bus and data bus; $\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$; RST 5.5, RST 6.5 and RST 7.5 interrupts; system clock, reset out and reset in. The programmable HDLC/SDLC protocol controller (INTEL 8273) has been used for the link interface to support some lower level data link controls. It minimizes the CPU software by supporting a comprehensive frame-level instruction set by hardware implementation of the low level real time task associated with frame assembly/disassembly and data integrity etc. The work

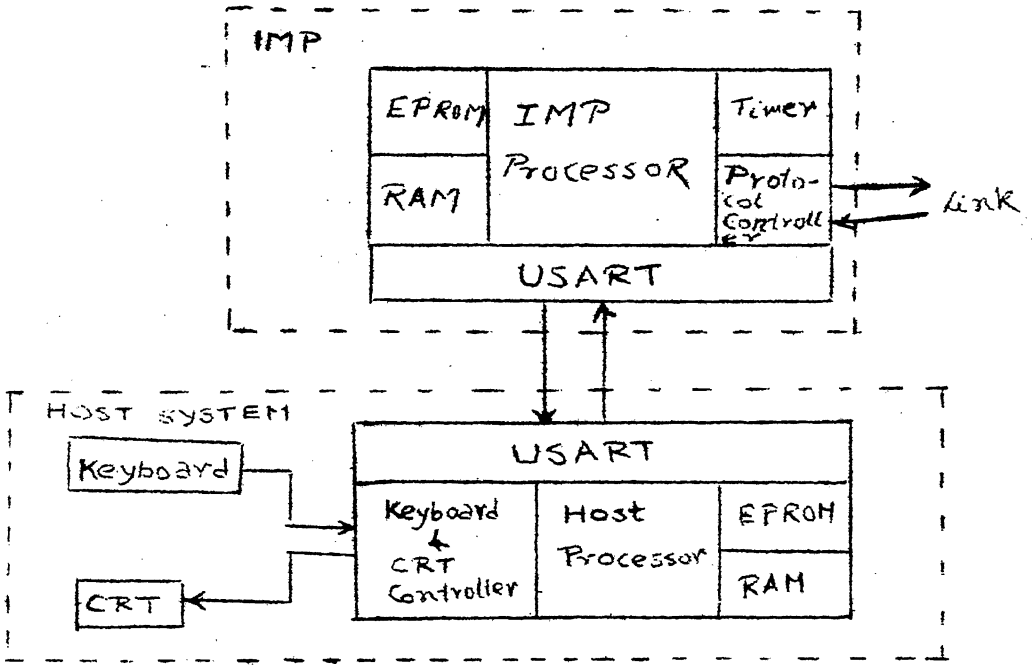


Fig 3.1 Block Diagram of the IMP & THE HOST

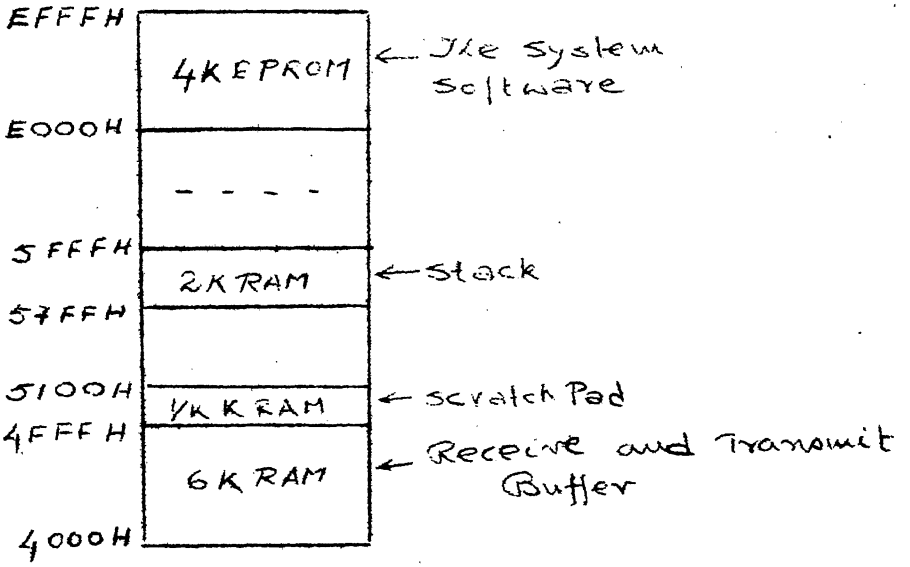


Fig 3.2 Memory Map of the System

station has 8K bytes of RAM (16x2114) occupying the address space 4000 H - 5FFFH. Of which 4000H-4FFFH has been used for transmit and receive buffer. 5000-50FFH has been used as scratch pad. 5800-5FFF are used for stack of the system. The memory organisation is shown in Fig. 3.2.

I/O card of the WORK STATION has got the programmeable timer (8253) which has been used for the time out operation. The output of counter 2, brought out through the connector is connected to RST 6.5 interrupt of the processor. The Monitor card which has an ASCII keyboard and the video monitor is controlled by a seperate 8085 processor. It reads characters from the keyboard, sends them serially to the 8085 of the processor card(IMP processor) through an USART. It displays the keyed in characters on the TTY. The characters to be displayed from the IMP processor comes through the serial link to the monitor card processor. It has its own 2K RAM for storing the characters constituting a 24 rows by 80 characters page displayed on the CRT screen. The monitor card with the keyboard and TTY constitute the Host system of the Interface Message Processor.

Fig. 3.3 gives a schematic of the Interface Message processor. The main chips used and the functions performed are described below.

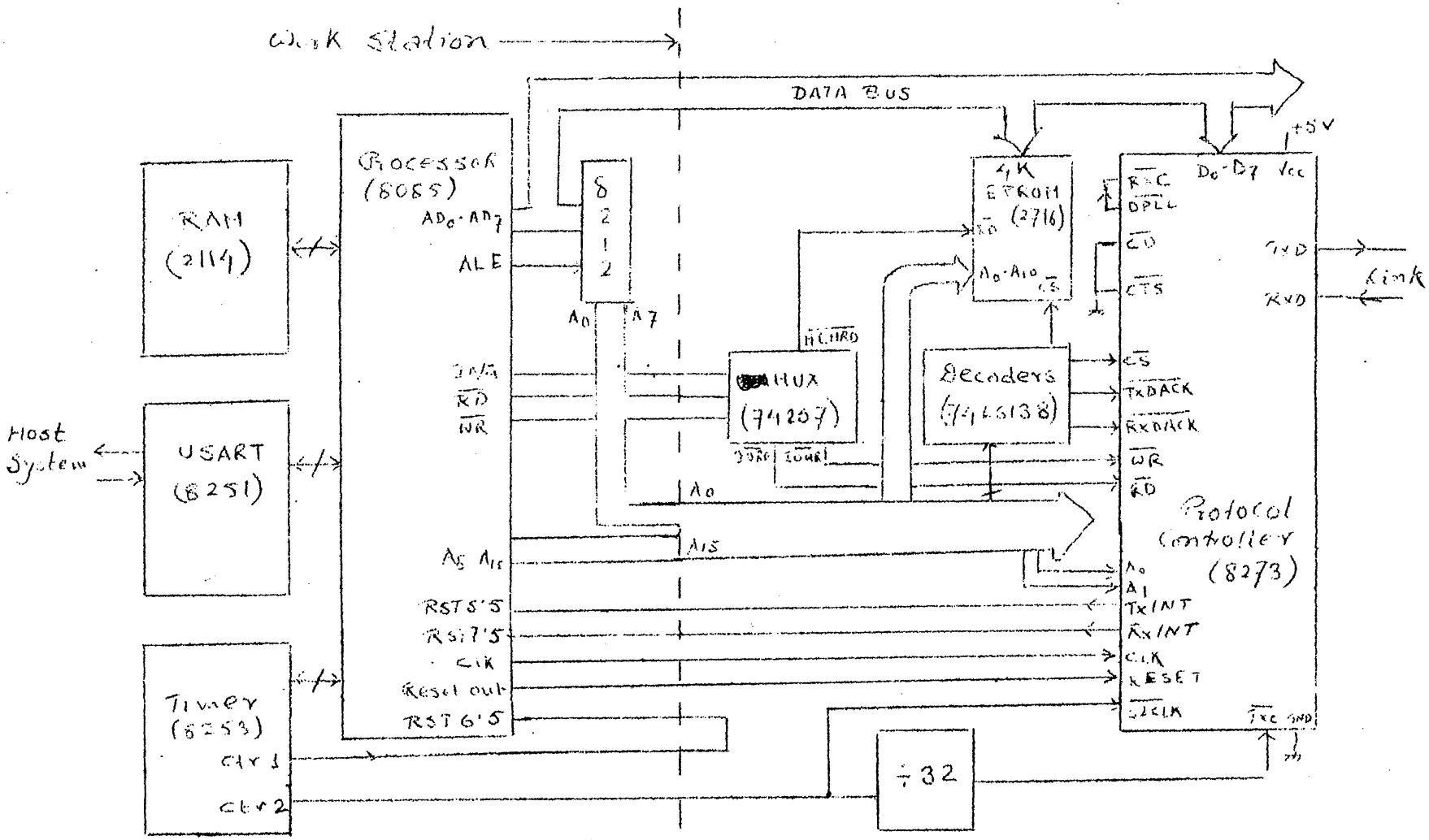


Fig 3.3 Schematic Of the Interface Message Processor

3.1.1 The Programmable HDLC/SDLC Protocol Controller (8273):

Fig. 3.4 shows the functional block diagram of 8273. It has been used in asynchronous full duplex mode. The NRZI encoding/decoding of the receive and transmit data is used. The serial data is synchronized by the user transmit ($\overline{TXC}$) and receive ($\overline{RXC}$) clocks. The leading edge of $\overline{TXC}$ generates new transmit data and the trailing edge of $\overline{RXC}$ is used to capture receive data. The receive clock is derived from the NRZI data using a digital phase locked loop (DPLL). As the data transparency is achieved by using a zero bit insertion technique as described in Sec.2.3.2, the use of NRZI coding guarantees a data transition at least in every five bit times. $\overline{DPLL}$ is connected to receive clock (RXC). The $\overline{DPLL}$ output requires two inputs. A clock at 32 times the required baud rate and Receive Data (RXD). But there are two problems associated with it. i) Initially, 8273 cannot know whether the $\overline{DPLL}$ is correctly synchronized with the received serial data bit stream. ii) The 32-Clk may not be exactly 32 times the serial data baud rate. The $\overline{DPLL}$ synchronization logic resolves these problems by using every active transition of RxD. The synchronization logic divides serial bit time into four quarters; the width of each is eight clock pulses. This is shown in Fig. 3.5. Where the $\overline{DPLL}$ is correctly synchronized

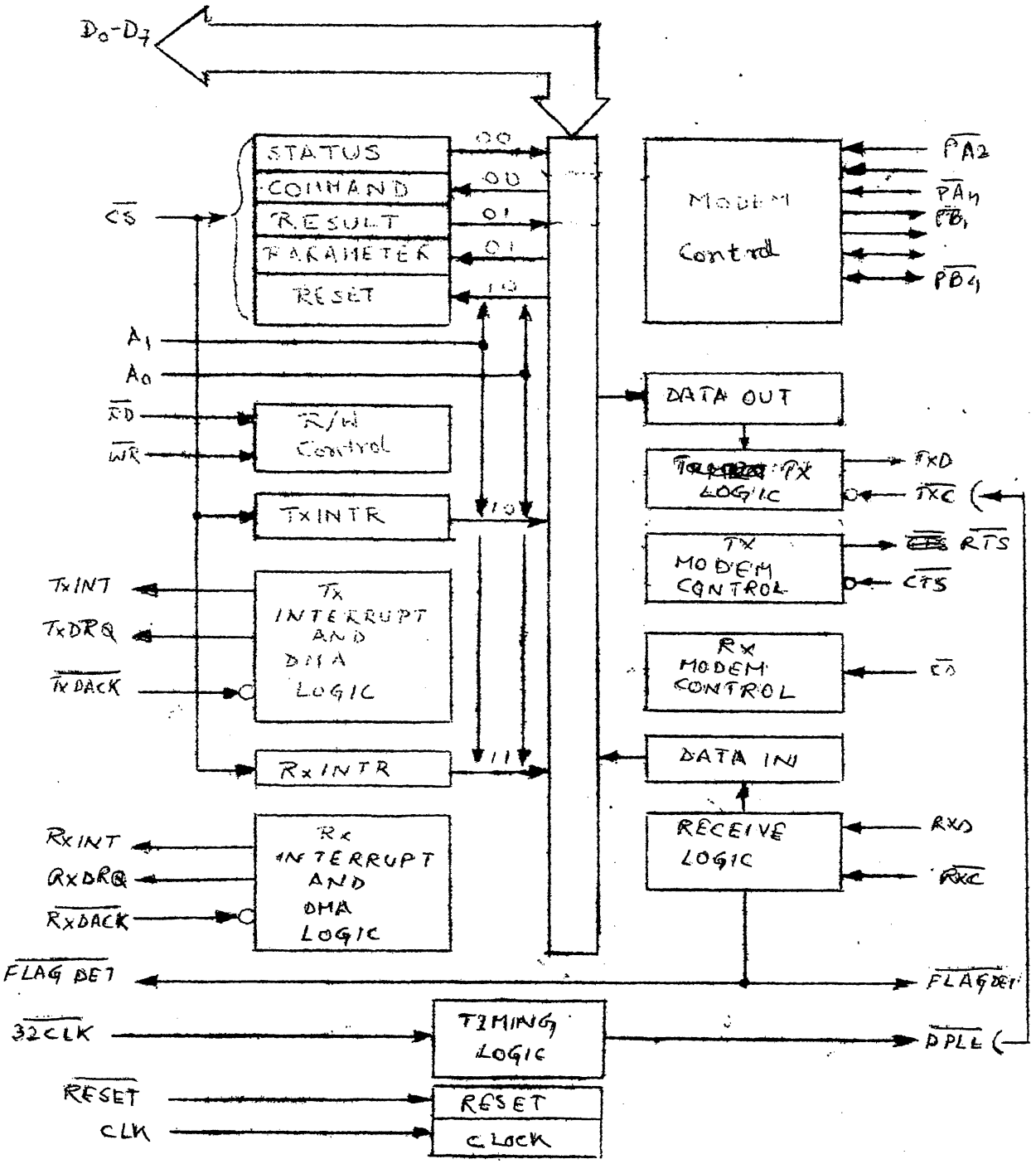


Fig 3.4 8273 Functional Logic

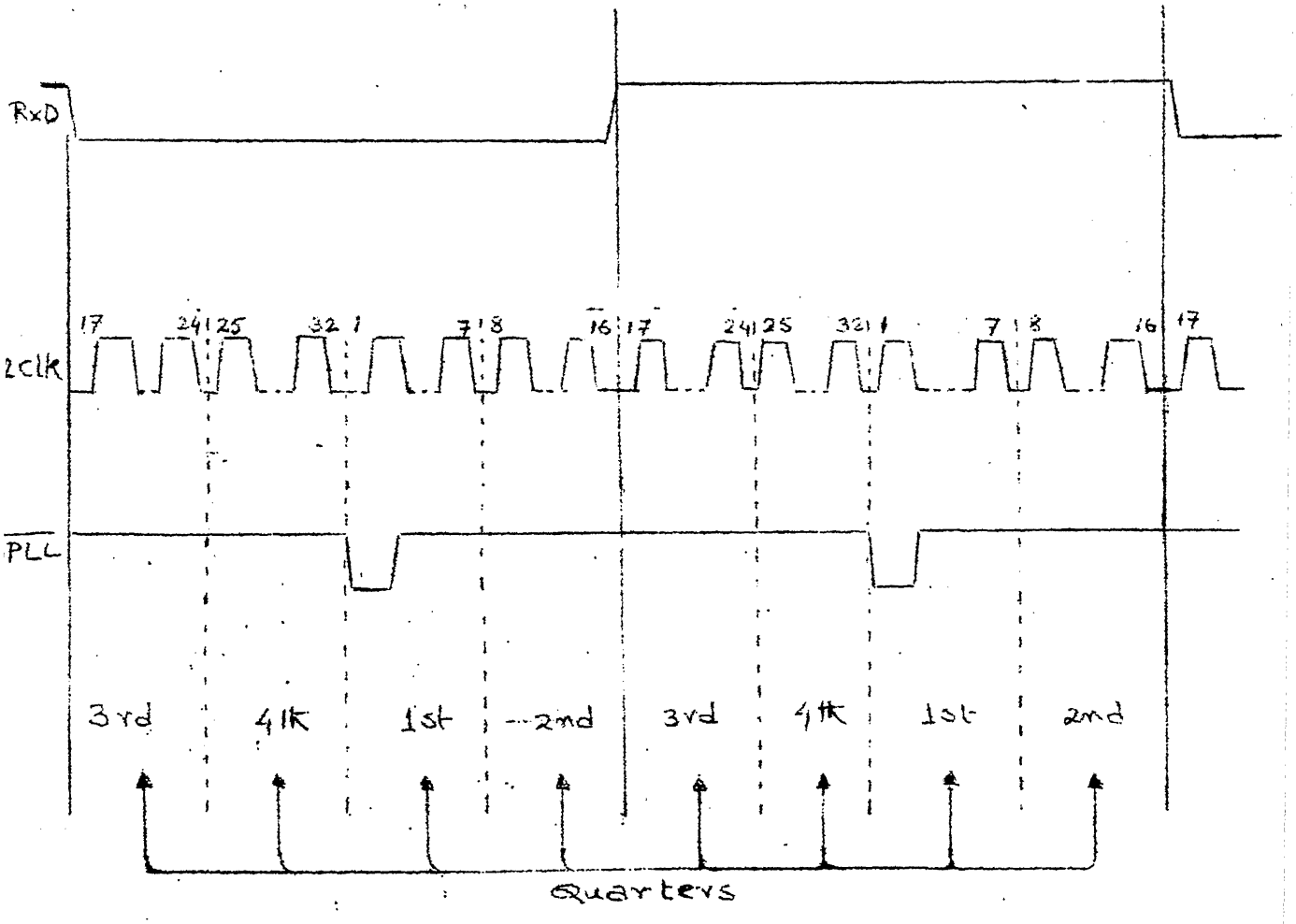


Fig 3.5 DPLL Clock Adjustment

RxD transitions will occur between the second and third quarters of 32 Clk. As $\overline{DPLL}$ transition always occurs at the end of 4th quarters, it will fall at the middle of RxD. Now if the RxD transition occurs at 1st quarter the $\overline{DPLL}$ is advanced in steps of 2 pulses. If it is in the 2nd quarter it is advanced in steps of 1 pulses. If RxD makes active transition in 3rd quarters $\overline{DPLL}$ retards by one pulse and two pulses for the transition in 4th quarters. So in worst case 12 transitions are needed for synchronization. This is achieved by programming 8273 in preframe sync. mode. By which 16 signal transitions are transmitted before initiating any frame tranmission.

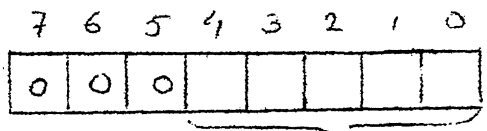
The 8273 appends the beginning and the ending flags $(7E)_H$. It aids in achieving data transparency by zero bit insertion and stripping it off from the received data. It adds CRC-16 at the end of a frame before the final flag is transmitted. The frames are automatically checked for errors during reception by verifying the CRC-16. It has been programmed in the interrupt driven data transfer mode. For transmit logic, the no. of bytes to be transmitted is specified. For each byte it generates interrupt through TxINT line. The data is written into the Data out register by pulling the

$\overline{\text{TxDACK}}$ line low. RST 5.5 is connected to TxINT line. At the end of transmission it generates one end of frame interrupt to give the interrupt result. The transmit interrupt result byte is interpreted as in Fig. 3.6.

After assembling the incoming serial data into the buffer it generates receive interrupt through RxINT line. RST 7.5 is used for receive interrupt. The data is read from the Data-In register by pulling the $\overline{\text{RxDACK}}$ line low. It has been programmed in unbuffered mode. So the receive interrupt result byte is to be read thrice. The first byte gives the receive result. The byte is interpreted as in Fig. 3.7. The 2nd and the 3rd bytes give the number of bytes received. The 8273 has been programmed in general receive mode.

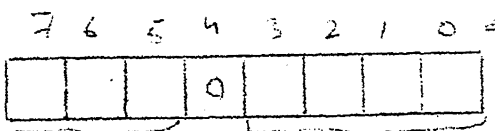
3.1.2 The Timer (8253):

The counter 0 and counter 1 are used for time out. The minimum timeout time should be greater than the message processing time + time to initiate ack + time to process the ack on reception. So the minimum time out time = 0.75 msec. The counter 1 is programmed in MODE 3 to generate square wave of lower [illegible] frequency. This clock is fed to the counter 0 clock. The counter 0 is programmed in MODE 0 to get an interrupt on terminal count. RST 6.5 interrupt is used for time-out interrupt.



00000
⋮ Unassigned
01011
01100 Early transmit interrupt
01101 End of flag frame transmit complete interrupt
01110 DMA underrun error interrup
01111 Clear to send error interrupt
10000 Transmit about completed
10001
⋮ Unassigned
11111

Fig 3.6 Tx Interrupt Result



0 D0
1 D4-D0
0 D2-D0
1 D6-D0
0 D1-D0
1 D5-D0
0 D3-D0
1 D7-D0

0000 A1 match or general receive
0001 A2 match
0010 Not assigned
0011 CRC error
0100 Abort detected
0101 Idle detect
0110 EOP detect
0111 Frame less than 32 bits
1000 DMA overrun error
1001 Memory buffer overflow
1010 Carrier detect failure
1011 Receive interrupt overrun
1100
⋮ Not assigned
1111

Fig 3.7 Receive Interrupt Result

3.1.3 The EPROM (2716):

4K system software is placed in the address space E000H to E7FFH using two EPROMS (2716). The address decoding logic for the address space OE000H to OE7FFH is $\overline{CS}_1 = \overline{A_{15} \cdot A_{14} \cdot A_{13} \cdot \bar{A}_{12} \cdot \bar{A}_{11} \cdot \overline{IO/M}}$ and for OE800H to OEFFFH is $CS_2 = A_{15} \cdot A_{14} \cdot A_{13} \cdot \bar{A}_{12} \cdot A_{11} \overline{IO/M}$. The decoding is performed using 74LS138.

3.1.4 The USART (8251):

The IMP is interfaced with the host system through a serial link using 8251. It is programmed in the asynchronous mode with transmit and receive baud rate 2.4K baud. The CPU looks at the RxRDY bit of the status register for any keyed in character from the host. To display some message into TTY the CPU polls the TxRDY bit of status register and sends it via the serial link.

The Fig. A.1. in Appendix A gives the details circuit diagram. The registers and their port numbers are given in Table 3.1.

3.2 DESCRIPTION OF SOFTWARE:

It has been assumed that all the nodes have equal status i.e. every one needs to have the ability to initiate link connection and disconnection. So the nodes are in Asynchronous Balance Mode.

Table 3.1

Device	Register	Port No.	Read/write location
8273	Command	10H	W
	Parameter	11H	W
	Status	10H	R
	Result	11H	R
	Transmit Interrupt	12H	R
	Receive Interrupt	13H	R
	Reset	12H	W
	Data In	20H	R
	Data Out	30H	W
8253	Control Word	C3H	W
	Counter 0	C0H	W
	Counter 1	C1H	W
	Counter 2	C2H	W
8251	Control	F9H	W
	Status	F9H	R
	Data in	F8H	R
	Data out	F8H	W

3.2.1 Command and Responses:

The following commands or responses have been implemented.

Format	Control bytes	Acronyme	Function	Name
U	1111x100	SABM	C	Set asynchronous balance mode
	1100x010	DISC	C	Disconnect
	1100x110	UA	R	Unnumbered acknowledgement
S	$1000xN_r$	RR	R	Ready to receive
I	$0N_s$xxxx	I	C	Sequenced I-frame

X = unspecified, U= unnumbered, S= supervisory,

I = information, C = command, R = response

SABM (Set Asynchronous Balance Mode): This command sets the logical link between two stations. UA is the expected response.

DISC(Disconnect): This command terminates other station's logical link. The expected response is UA.

UA (Unnumbered Acknowledgement): This is the affirmative response to a SABM and DISC command.

RR (Receive Ready): Sent by a station to confirm the numbered

frames through N_r and indicates that the originating station is ready to receive. This is the acknowledgement frame.

I (Information): These frames are numbered. The N_s count provides the numbering the frame being sent. It expects acknowledgement through RR frames.

3.2.2 Data Structure:

For implementation the data structure chosen for all the buffers are cyclic queues. When a packet is put in a receive or transmit queue, it is not itself transferred. Only the rear and front gets changed to access that packet. Thus without much data transfer the required functions can be performed efficiently.

The transmit buffer can keep eight packets (Fig. 3.8). Transmit Buffer Front (TFRNT) points to the packet which has to be transmitted. Transmit Buffer Rear (TREAR) points the location where data will be loaded from host system. State of Transmit Buffer (SOTB) gives the present status of transmit buffer. Transmit Frame Sequence (TFSQ) keeps track of the sequence no. of the transmitted packet.

A buffer is maintained to keep the number of bytes in each packet which is needed to programme the 8273 in transmit mode.

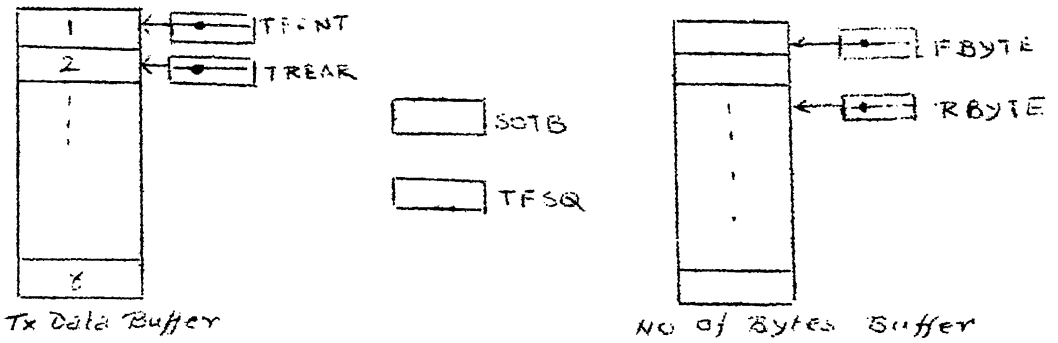


Fig 3.8 TX Buffers

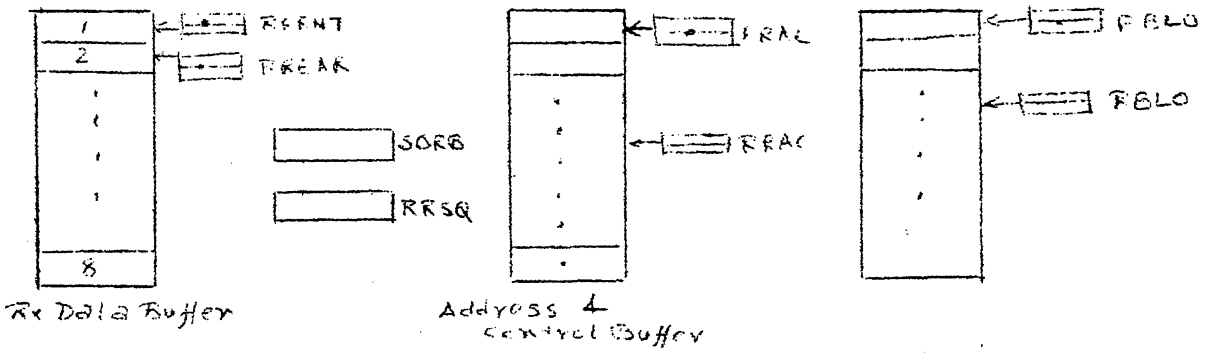


Fig 3.9 Rx Buffers

FL2 => 0 for address, 1 for control, 2 for Data

SAV1 => 0 to load data in Address and Control Buffer

1 to load data in Rx Data Buffer.

FL7 => Bit 7 is 0 token timer disabled

Bit 7 is 1 and bit 0 is 0 for the timeout of I-frames

Bit 7 is 1 and bit 0 is 1 for the timeout of SABM

Bit 7 is 1 and bit 0 is 1 and bit 1 is 1 for the timeout of DISC

Fig 3.10 Flags referred in the given flows charts

Similarly for the received data, there are eight buffers. Receive Front (RFRNT) points to the location from which the data has to be displayed. Receive Rear (RREAR) points to the location where the received data has to be has loaded. State of Receive Buffer (SORB) gives the present status of receive buffer. Receive Sequence Number (RRSQ) keeps track of the sequence number which has to be sent as the acknowledgement of the received information frame. One buffer has been maintained to keep the address and control bytes of the received frame which is necessary to take the required action on reception of a frame. No. of received bytes buffer is maintained to transfer data from IMP to host system.

3.2.3 Link set up and Close Down Phases:

On power on, the stations are in the disconnected state. To set up link, the station has to send the SABM command. The sender starts a timer as soon as he transmits an SABM and waits for an acknowledgement (UA) for the SABM. Sender may timeout and retransmit the frame if he does not get the UA within a reasonable length of time. To avoid an obvious infinite loop, it stops after retransmitting eight times and returns to the disconnected state. (This procedure is carried out for all frames needing an acknowledgement).

Once it receives an acknowledgement (UA) for the SABM, it enters the connected state where the actual information transfer can proceed. Whenever it enters the connected state, it sets the sequence no. of the frame TFSQ to O. On receipt of a SABM, the station initiates UA transmission. It also initializes its sequence no. RRSQ to O and it is ready to receive data.

The link disconnection phase is very similar to the above. If station wants to cutoff the link, it sends DISC command. On receipt of a DISC command it checks whether any frame is unacked. If all the frames are not acked the DISC remains unacked. The station goes off the air on reception of UA for DISC.

3.2.4 Data Transfer Phase:

On getting data from host in Tx Data Buffer the transmission is initiated. The address and control words are transmitted. After that data is transmitted from the location pointed by TFRNT. Timer for timeout is initiated. If no acknowledgement is received it retransmits the frame on timeout interrupt (RST 6.5) and goes back to wait mode. If it gets the acknowledgement TFRNT points to the next buffer to transmit the next frame.

On receiption of a frame address and control is loaded to the location pointed by RRAC, number of bytes received is loaded to the content of RBLO and the data is loaded to the buffer pointed by RREAR. The end of received frame result of 8273 is checked. If there be no error the control byte processing starts. If the N_r bits of control bytes is the expected seq. no., the acknowledgement transmission is initiated If the frame is damaged - all the pointers are restored to its previous locations.

So when I frame gets damaged, simply retransmission recovers the situation. When acknowledgement frame gets damaged the sender times out and retransmits. The receiver, finds the duplicated frame by checking the sequence no., rejects it and again sends the acknowledgement for the frame.

The timeout done is random. If the timeout time be constant, all the affected senders will retransmit simultaneously and will lead to the same signal to interference ratio. So the packets might get damaged again.

Refer Fig. 3.11, Fig. 3.12 and Fig. 3.13 for RST 5.5, RST 6.5 and RST 7.5 interrupt service subroutines. Fig. 3.14 shows the flow chart for the necessary decision taken on reception of a frame.

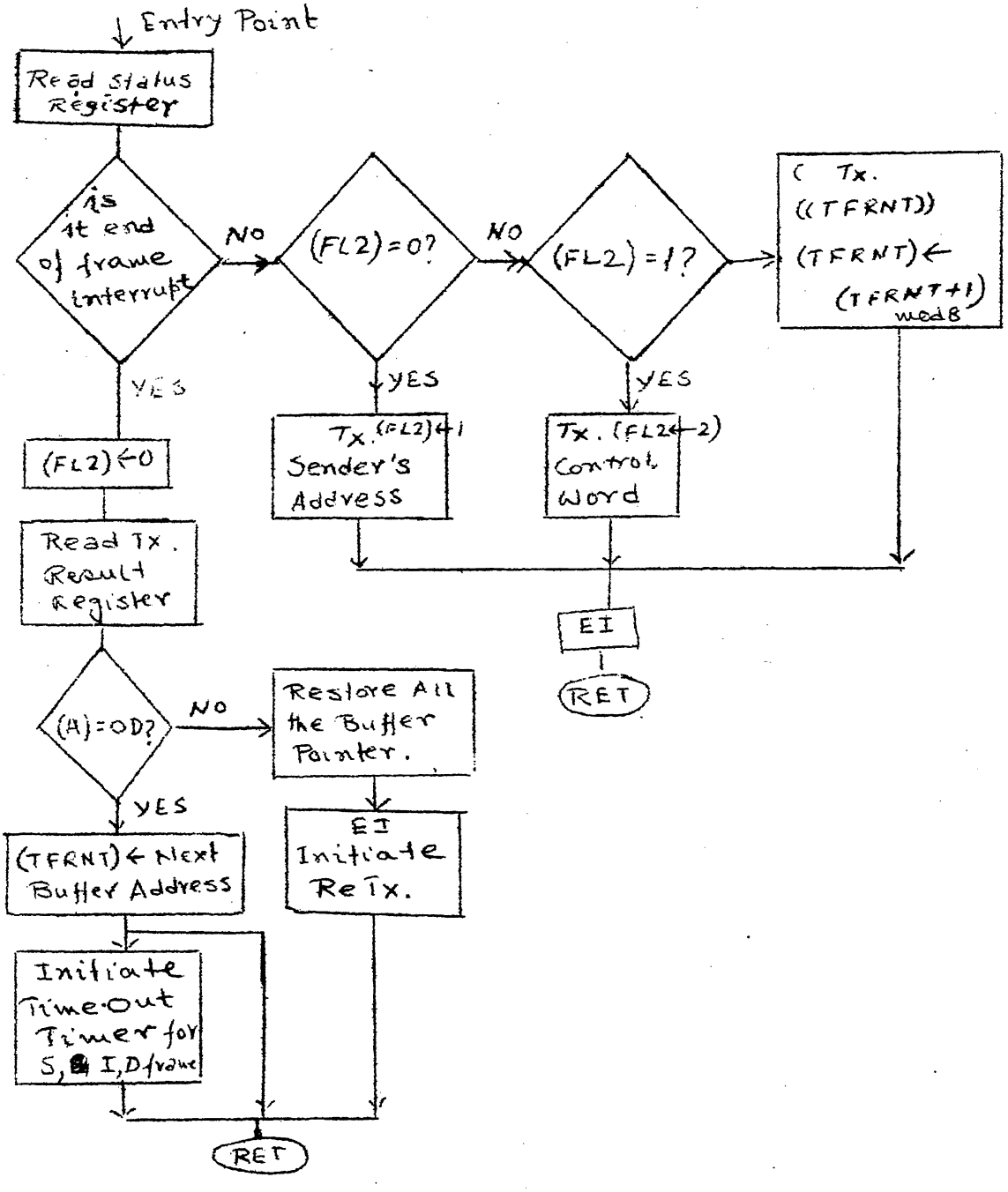


Fig 3.11 Tx mit Interrupt Subroutine (RST 5.5)

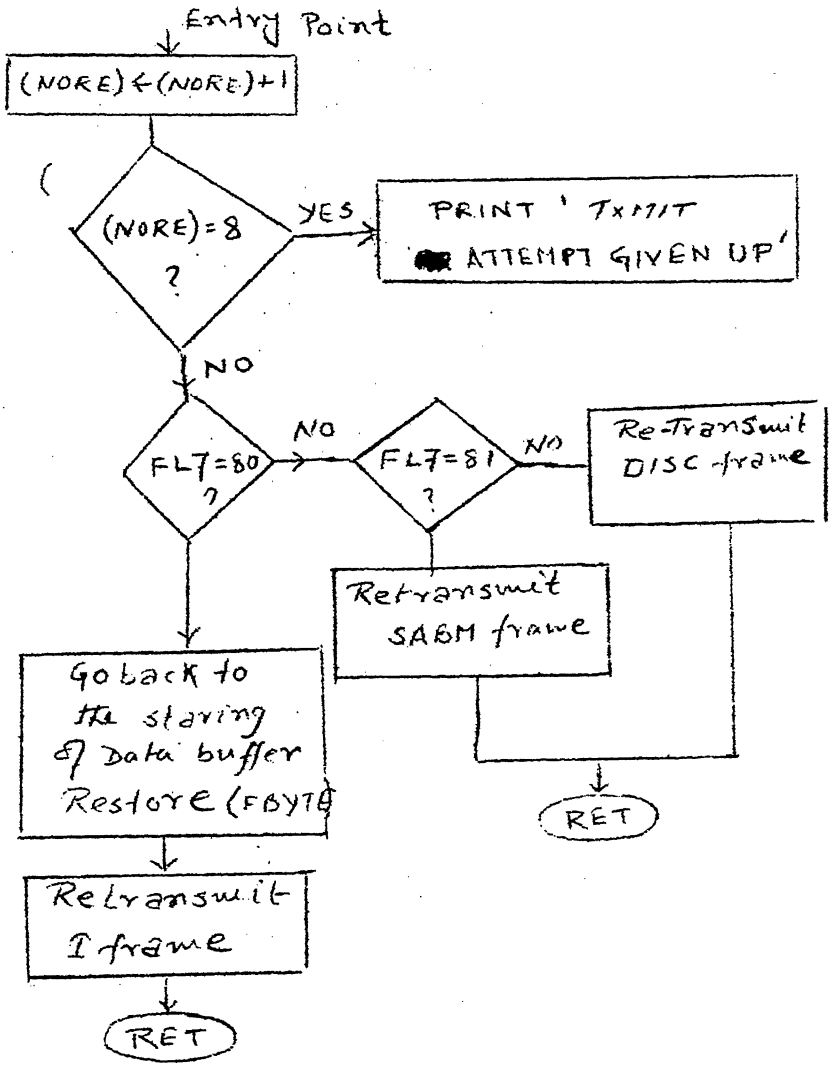


Fig 3.12 Time Out Interrupt (RST 6.5)

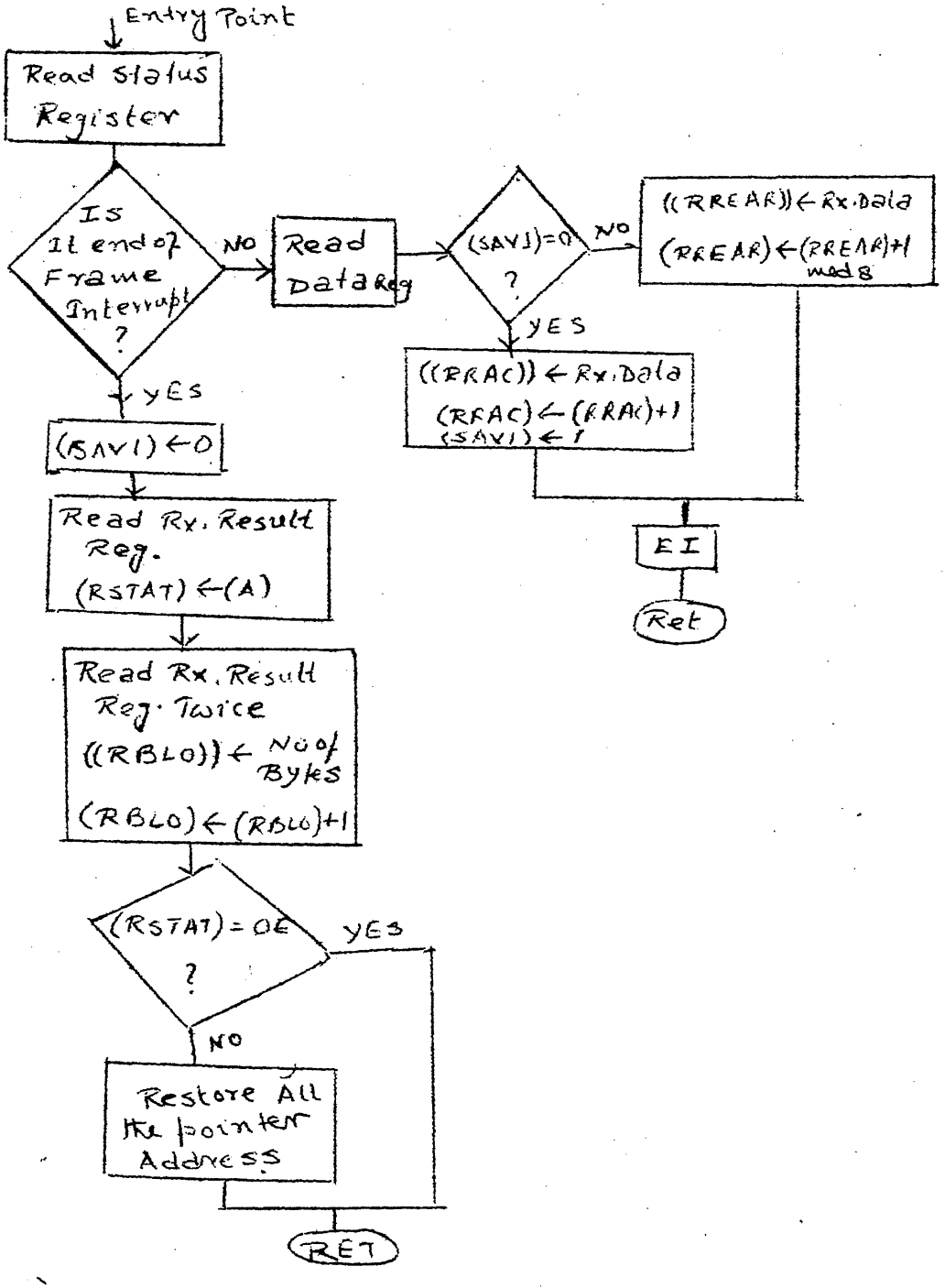


Fig 3.13 Receive Interrupt Subroutine (RST 7.5)

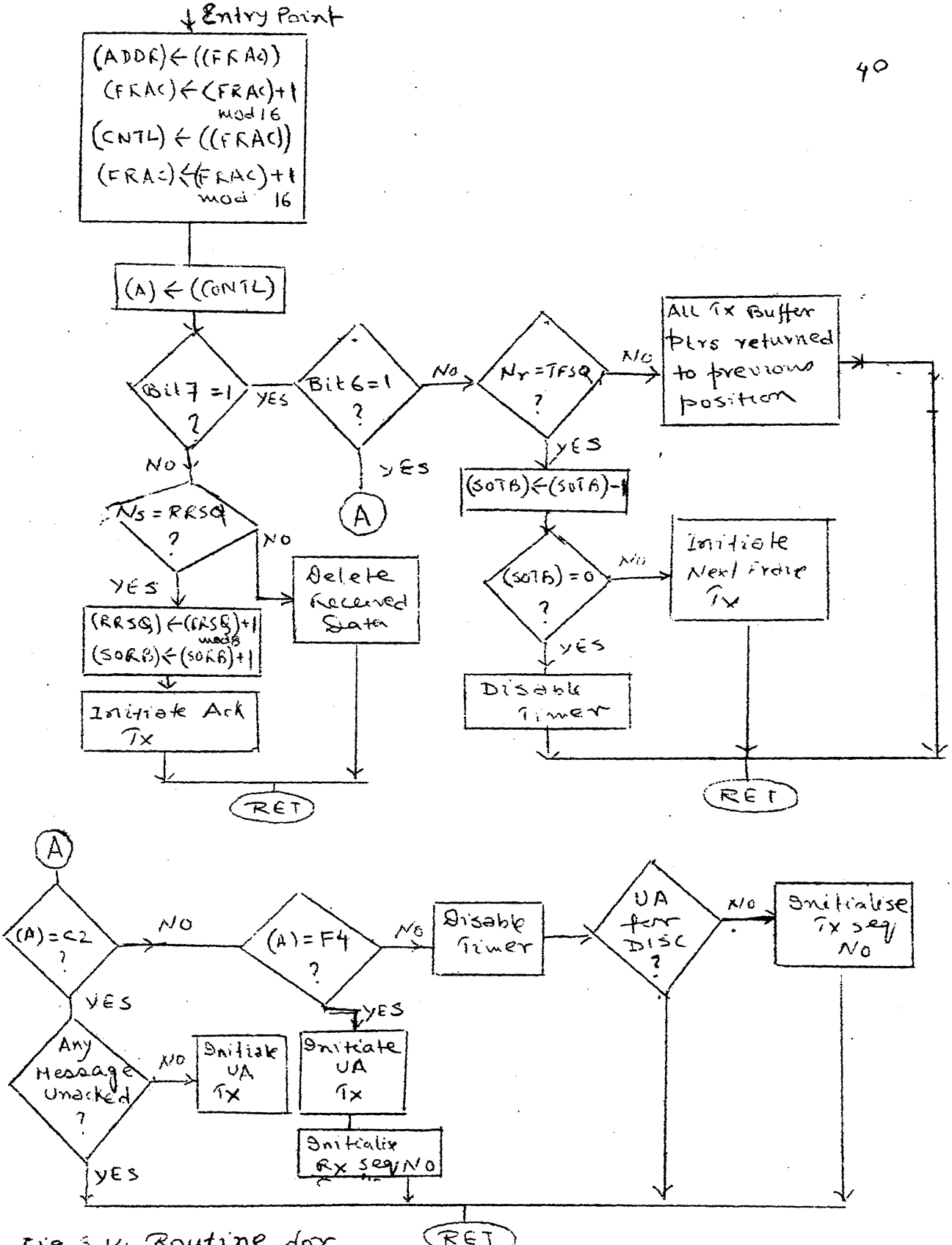


Fig 3.14 Routine for Decision after Receiving a Packet

3.2.5 Users' Interface:

The following routines were developed to implement the interactive features in the host system.

Name of the routine	Operation done
READ	Loads the keyed in ASCII character in Acc
PRINT	Displays on the TTY the ASCII character stored in the Acc
HEXBI	Converts the characters in B and C to form a byte available in B.
PNTMS	Prints a message on the TTY; BC points to the starting of the message which is ended by a '*'
CRLF	Gives a carriage return and line feed on the TTY.

The following comm have been implemented to interact through the host system.

The command 'L' is to load the keyed in data in memory. Refer Fig. 3.15 for it. It also packetizes the received data and arranges in buffer.

The command 'R' displays the received data. Refer Fig. 3.16 for the flow chart.

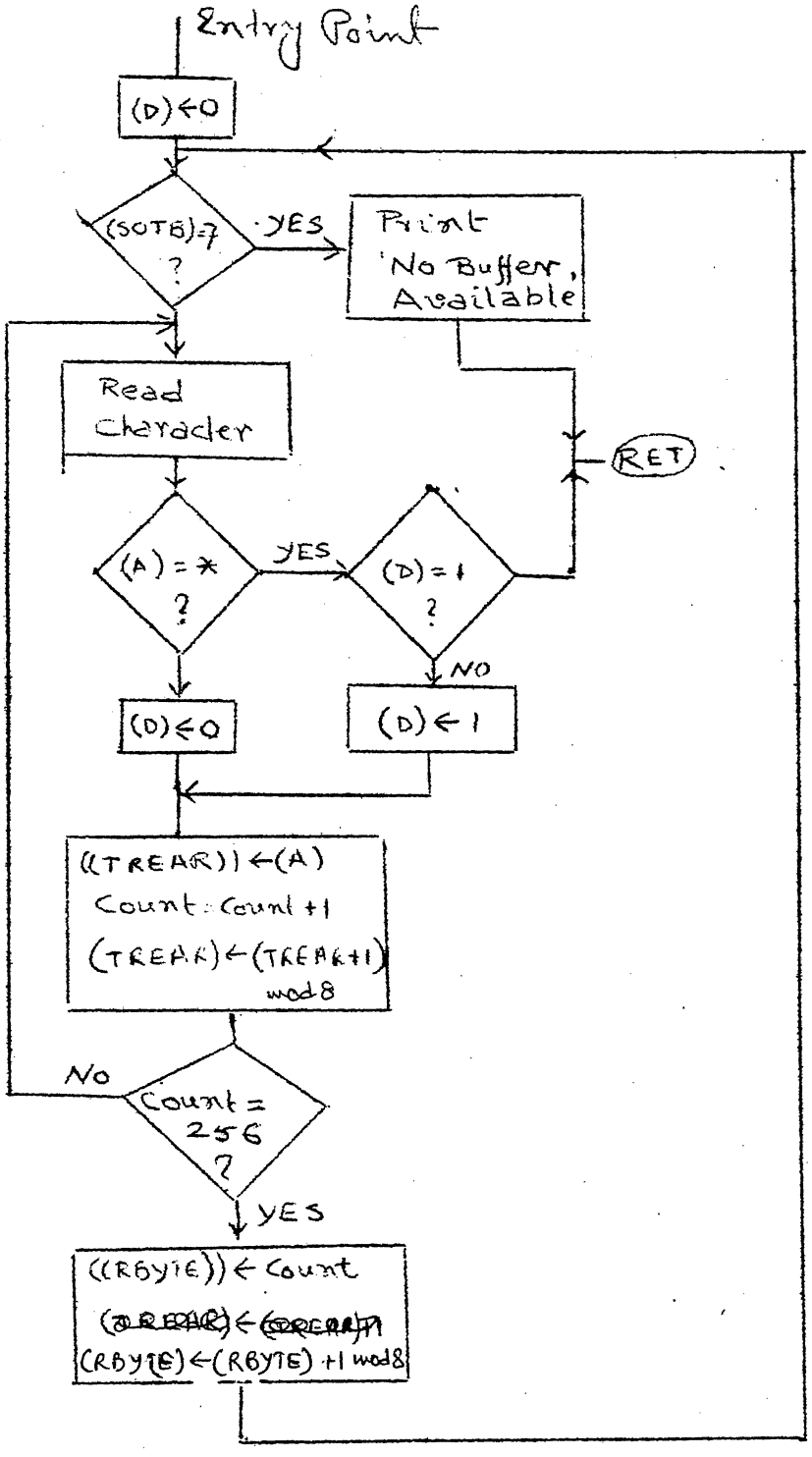


Fig 3.15 Routine after Pressing 'L'

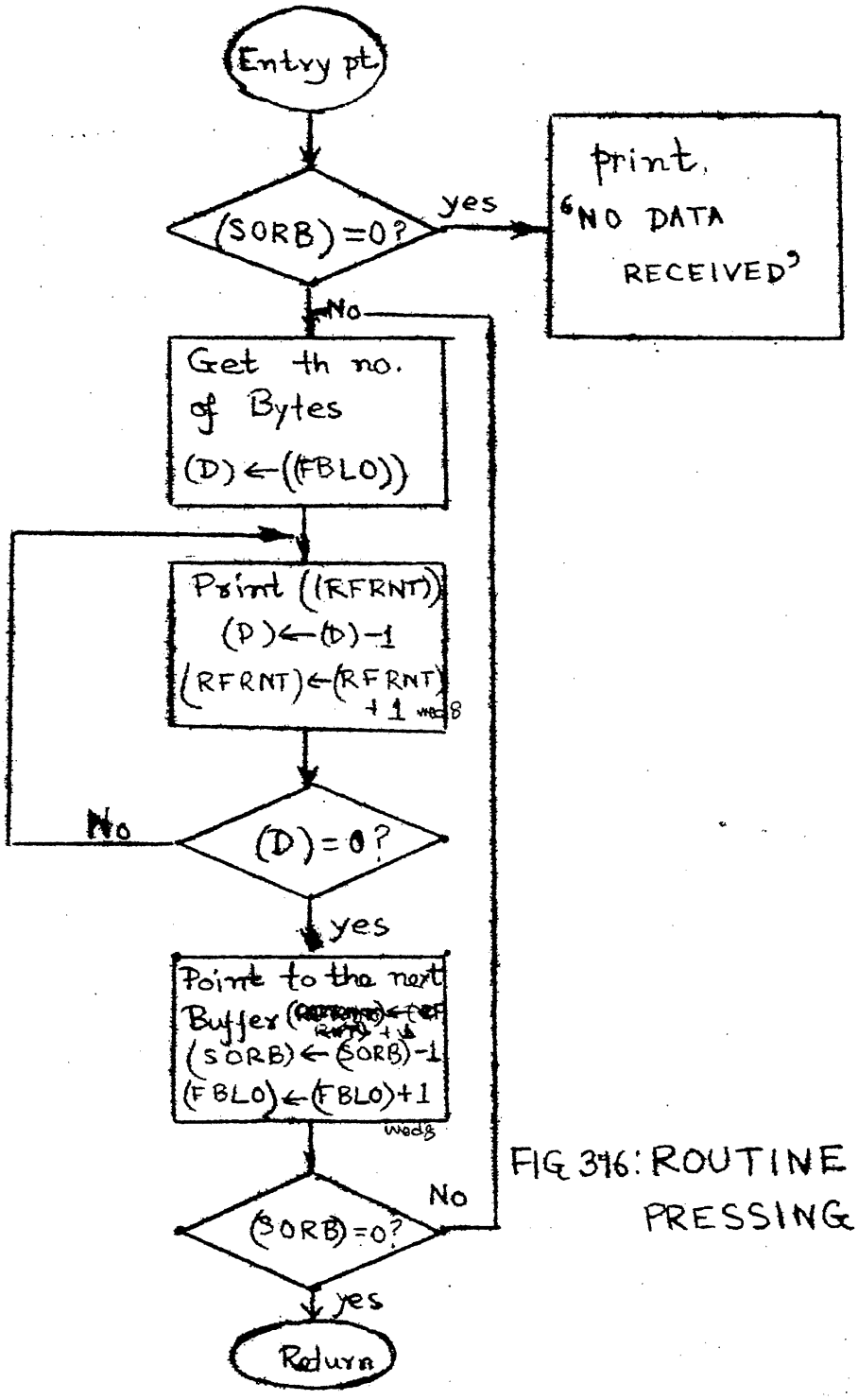


FIG 3.16: ROUTINE AFTER PRESSING R

The command 'S' initiates SABM transmission to set up the link. It asks for destination address. Depending on the address it enables the appropriate transmitter. This is performed by enabling/disabling a tristate buffer using the port B of the 8273 under program control.

The command 'T' initiates I frame transmission

The command 'D' is to disconnect the link

The command 'I' initializes all the buffers and resets the 8273.

CHAPTER 4

This chapter deals with the spread spectrum scheme adopted in the physical layer to improve the performance of the system in random access environment. It describes the spread spectrum scheme and its implementation details.

4.1 DIRECT-SEQUENCE SSMA COMMUNICATION:

In the binary direct-sequence SS modulation the spreading signal is of the form

$$a(t) = \sum_{j=-\alpha}^{\alpha} a_j P_{T_c}(t - jT_c)$$

a_j is a periodic sequence of elements $\{0, 1\}$

$$P_{T_c} = \begin{cases} 1 & , \ 0 \leq t < T_c \\ 0 & , \ \text{otherwise} \end{cases}$$

The binary data signal is given by

$$b(t) = \sum_{l=-\alpha}^{\alpha} b_l P_T(t - lT)$$

where P_T is the rectangular pulse of duration T which starts at t=o and $b_l \in \{0,1\}$ for each l. The baseband spread - spectrum signal is then $s(t) = a(t)b(t)$. The sequence a_j is called signature sequence, which satisfies $x_j = x_j+N$ for each j. N is an integer multiple of the period of signature sequence. The bandwidth of s(t) is N times the bandwidth of

b(t). In SSMA communication system (Fig. 4.1) there are K such signals which are simultaneously transmitted. So for the kth user the signal is given by

$$S_k(t) = A\, a_k(t)\, b_k(t) \qquad 1 \leq k < K \tag{4.1}$$

All the transmitters are not time synchronous. So the received signal in Fig. 4.1 is given by $r(t) = n(t) + \sum_{k=1}^{K} S_k(t - \tau_k)$ where n(t) is additive while Gaussian Noise with the spectral density $\frac{1}{2} N_o$, and τ_k is the time delay associated with the kth user.

The correlation receivers are used for despreading. So the output of the ith receiver is

$$Z_i = \int_o^T r(t)\, a_i(t)dt \quad ; \quad \tau_i = 0$$

$$= \eta_i + A \int_o^T b_i(t) + \sum_{\substack{k=1 \\ k\neq i}}^{K} A \int_o^T b_k(t - \tau_k)$$

$$a_k(t - \mathcal{T}_k)\, a_i(t)\, dt \tag{4.2}$$

where $\eta_i \int_o^T n(t) a_i(t)dt$ is due to the channel noise. The 2nd term is due to the ith signal. The final term is the multiple-access interference due to K-1 signals. This signal is spread over a wide-band. The interference due to it with the ith signal is very small. So within the limit of interference the ith signal can be recovered.

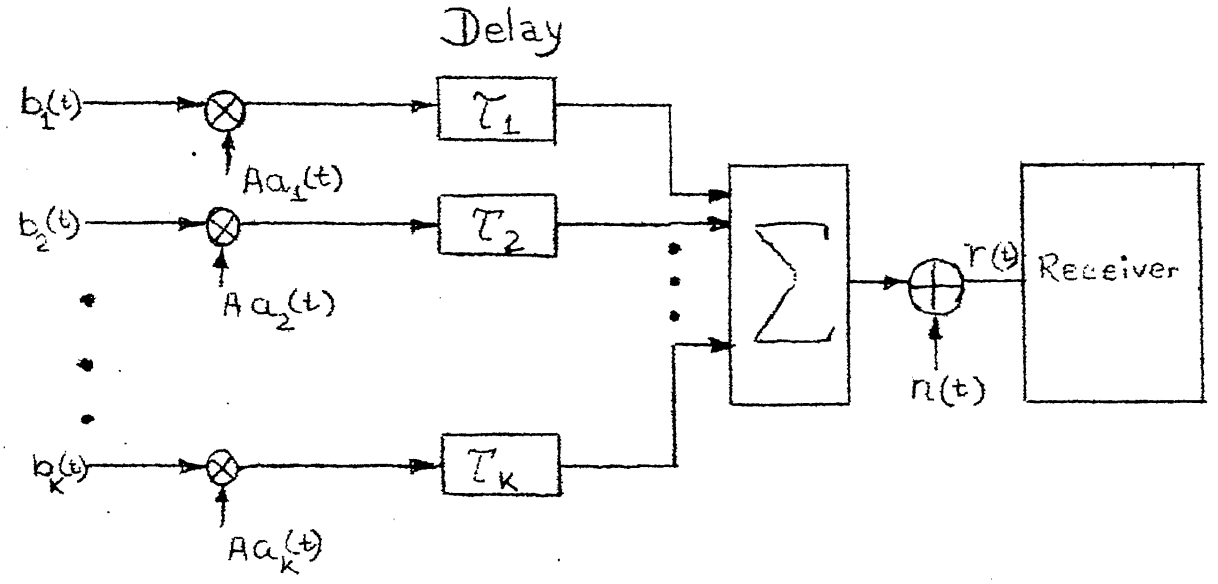


Figure 4·1 Binary Direct-Sequence SSMA Communication System Model

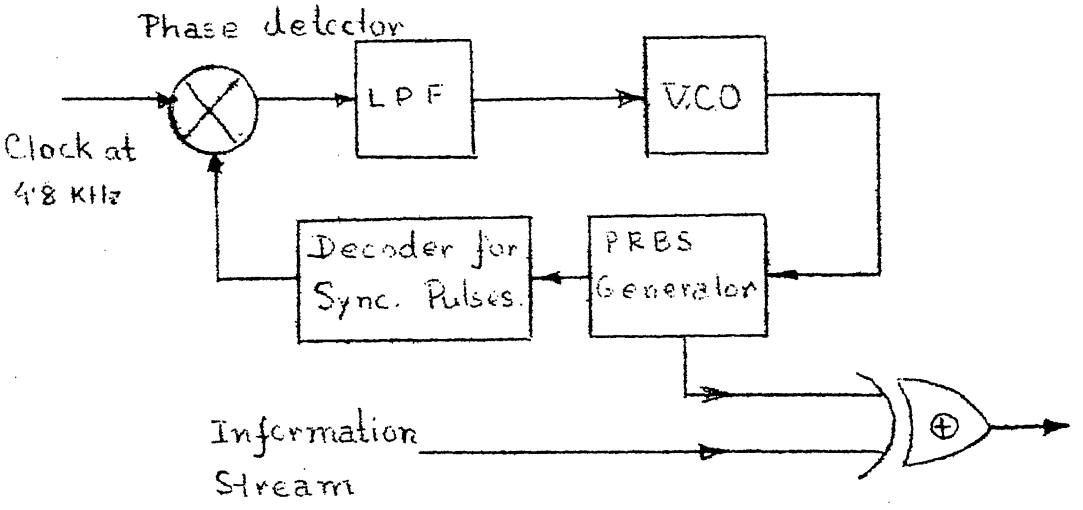


Figure 4·2 Schematic Block diagram of Transmitter

The maximal length PN sequence of 127 bit length has been used as spreading signal. As maximal length PN sequences are nearly orthogonal, the interference due to the other channels are quite low. Each subscriber is assigned a unique transmission code a_j in which all the messages addressed to it must be encoded. When a user u_k wishes to transmit a packet he unconditionally sends data encoded with code a_k.

The sections 4.2 and 4.3 describe the spread spectrum transmitter and receiver.

4.2 S.S. TRANSMITTER:

The main functions of the transmitter are:

i) Generation of 127 bit maximal length Pseudo-Random Bit Sequence (PRBS) in synchronism with the information signal.

ii) To modulate the data using direct sequence modulation.

4.2.1 Syncrhonous PRBS Generator:

The synchronous PRBS generator produces a 127 bit PRBS at 609.6 $\angle^{\text{Kbits/sec}}$ in synchronism with the input data stream. To generate the clock at 127 times the data clock, a phase locked loop has been incorporated in the subsystem. The data clock is at 4.8 KHz. The PLL contains phase detector (MC 4044), the low pass filter and the voltage controlled oscillator (MC 4024). To multiply the clock by 127 the output

of the shift-register stages are decoded by AND gates and fed to the phase detector input. The sync pulse is generated when all the shift register outputs are '1'. For the maximal length sequence generation, tappings three and seven have been EXORed and fed to the input of the shift register stages. If all the stages of the shift register enter zero state it cannot come out of it. Thus all zero prevention circuitry has been provided.

4.2.2 Modulator:

Modulator sub-system adds the information signal and the output of the 7th stage of the PRBS generator on the modulo-2 basis. Thus for each bit of information signal the outpput of the modulator sends 127 bit sequences.

The output of the 4th stage of the PRBS generator is ORed with the modulated data and transmitted. The bit shifted PN sequence is send to achieve the synchronization at the receiver. Fig. 4.2 gives a block schematic of the transmitter. The details circuit diagram is in Appendix A, Fig. A.2.

4.3 S.S. RECEIVER:

In order to recover the original information signal, the receiver generates a 127 bit PRBS in synchronism with the received bit stream. The locally generated code sequence has

to be synchronized both in frame and in bit. This code is the key to despreading the desired information and to spreading any undesired signals. Moreover, after initial acquisition the locally generated code sequence must continue to remain locked with the coded incoming signal; i.e. once the code sequences are matched in time, the receiver must cause its own code bit rate to match the incoming code bit rate within a small fraction of a clock period. The delay-locked loop has been implemented to achieve the synchronization. The received coded data is EXORed with the synchronized PRBS. After despreading the incoming data, it is integrated over a data period. By properly setting the comparator threshold the data is recovered. For the tasks to be accomplished, the receiver has the following sections.

i) Delay-Lock Tracking [21]

ii) Data Recovery

The receiver block schematic is shown in Fig. 4.3.

4.3.1 Delay Lock Tracking:

The Delay-Lock Loop (DLL) is a nonlinear feedback system, which employs a form of cross-correlation in the feedback loop. It is a tracking device for estimating the delay of any signal $S(t+\tau)$. For small values of delay

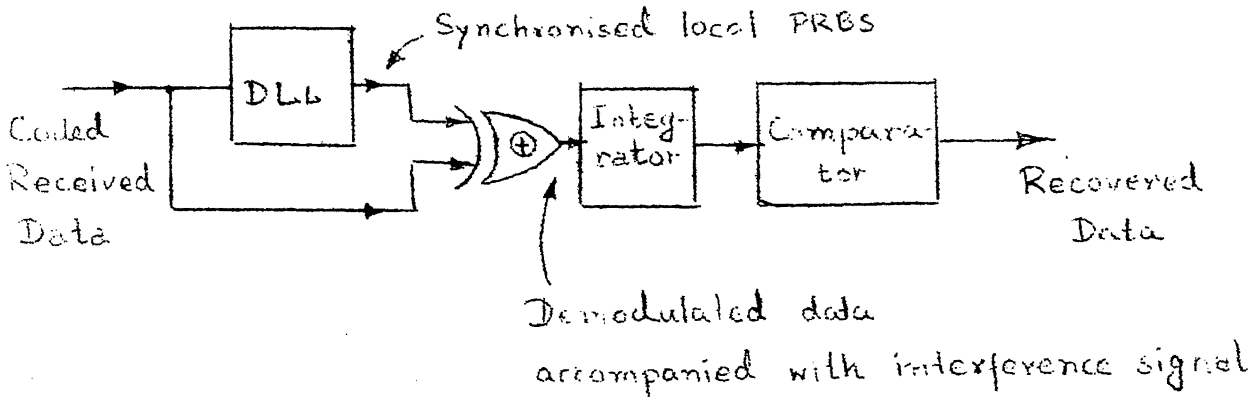


Figure 4·3 Receiver Block Schematic.

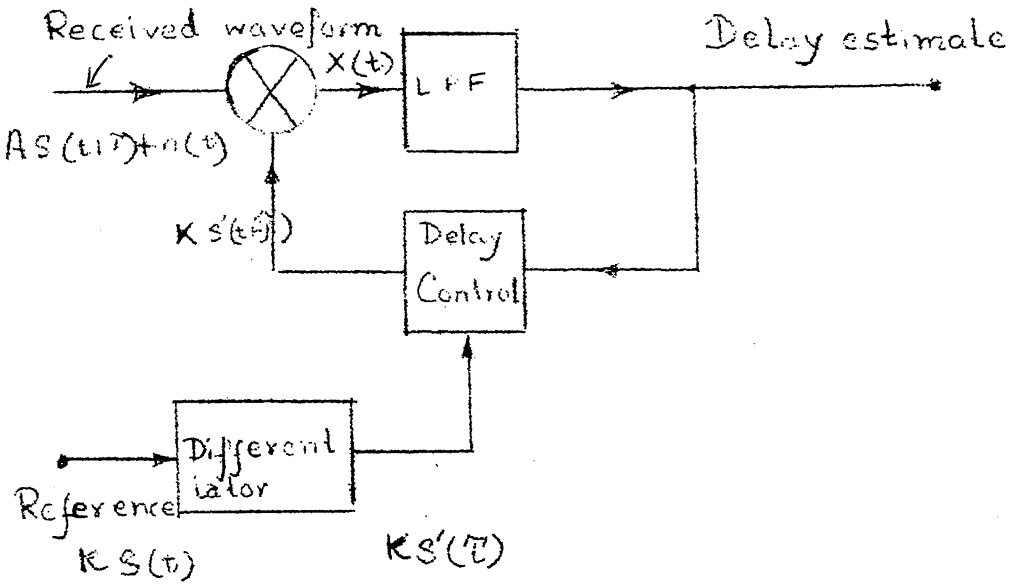


Figure 4·4 Functional Block Diagram of DLL

error, the loop provides an estimate of delay in the presence of Gaussian noise. The delay-lock loop is shown in Fig. 4.4.

Let the delay error be $\varepsilon(t) = \tau(t) - \hat{\tau}(t)$. Writing the Taylor's series expansion for the delayed received signal

$$S(t+\tau) = S(t+\hat{\tau}) + \varepsilon S'(t+\hat{\tau}) + \frac{\varepsilon^2}{2} S''(t+\hat{\tau}) + \ldots$$

where the primes refer to differentiation w.r.t. the argument, and all derivatives of $S(t)$ are assumed to exist. The multiplier output then has the series expansion

$$\frac{x(t)}{k} = A\left[S(t+\hat{\tau})S'(t+\hat{\tau}) + \varepsilon(t)[S'(t+\hat{\tau})]^2 + \frac{\varepsilon^2(t)}{2} S''(t+\hat{\tau})S'(t+\hat{\tau}) + \ldots\right] + n(t)S'(t+\hat{\tau}) \qquad (4.3)$$

Now if $S(t)$ is a PN sequence, then $S'(t)$ is a sequence of impulses of the same sign as the transitions in the PN sequence. $S(t)$ is normalized to have unity power, and thus the received signal power is $P_s = A^2$. In eqn. (4.3) the terms $(S')^2$ has a nonzero average value which is defined as P_d, the power in the differentiated signal. We can write (4.3) $[S'(t)]^2 = P_d + S_o(t)$ where $S_o(t)$ has zero average value. Now (4.3) can be rewritten as

$$\frac{x(t)}{k} = A P_d \varepsilon(t) + \eta_e(t) \qquad (4.4)$$

where the first term is the desired error correcting term, and the second term $\eta_e(t)$ is an equivalent noise term caused by the interfering noise $n(t)$ and the remainder of the infinite series, distortion and self-noise effects. If ε is small, $n_e(t)$ has little dependence upon $\varepsilon(t)$. From eqn. (4.4) the tracking behaviour can be explained as, the input delay $\tau(t)$ is suddenly increased by a small amount, the error $\epsilon(t)$ will also increase suddenly; the multiplier output will increase, and therefore the delay estimate $\hat{\tau}(t)$ will increase and tend to track the input delay.

Slight variation of the above described delay lock technique is implemented for the binary PN sequence.

Figure 4.5 gives a block diagram of the baseband binary delay lock loop. Referring to the Fig. 4.5.

the Δ is the bit width of the PN sequence. The binary signal, plus additive white noise is fed into the cross-correlation network, where it is compared with time-displaced versions of the same PN sequence as used in the transmitter. The correlation network has an output

$$k\delta S(t+\tau)[AS(t+\tau)+n(t)]$$

$$= [AS(t+\tau)+n(t)]\ k[S(t+\Delta+\hat{\tau})-S(t-\Delta+\hat{\tau})] \qquad (4.5)$$

$$\therefore \delta S(t) = S(t+\Delta) - S(t-\Delta)$$

$$\frac{dS}{dt} = \lim_{\Delta \to 0} \frac{S(t+\Delta)-S(t-\Delta)}{2\Delta}$$

It is the reminiscent of the expression for the differentiated signal which is used in eqn. (4.3).

The product contains a low-pass spectral component that serves to keep the DLL accurately tracking the system delay once the system has locked-on. The filter is to remove as much noise and other interference as possible and provide the desired closed-loop response.

From the equation (4.5) the DLL characteristic is given by

$$kA[D(\varepsilon)]$$

where $\varepsilon = \tau - \hat{\tau}$ is the delay error. $D(\varepsilon)$ is the DLL characteristic. $\mathcal{A}$ is the time average of the cross-correlation network for fixed ϵ.

So $$D(\varepsilon) = E[\delta S(t+\hat{\tau}) S(t+\tau)]$$

$$= R_s(\varepsilon-\Delta) - R_s(\varepsilon+\Delta) \qquad (4.6)$$

where operator E represents the ensemble average for a sequence S(t). From eqn. (4.6) the delay lock loop characteristic is the difference of the correlation of the incoming signal with the locally generated signal in two correlators with two bit-shift. The correlation function is

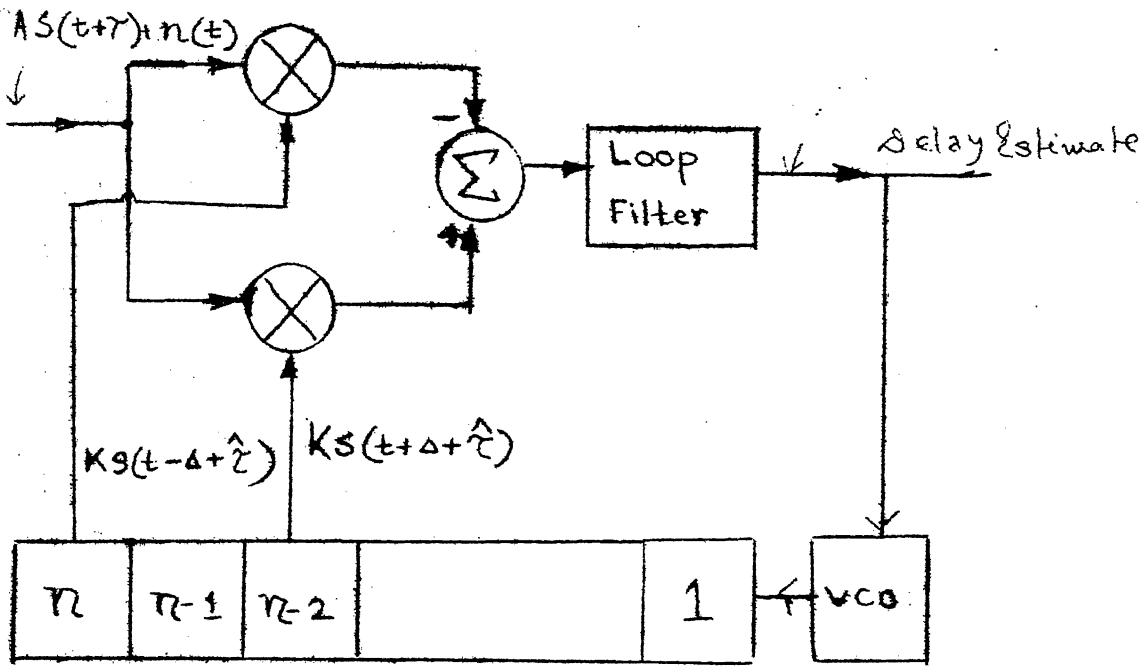
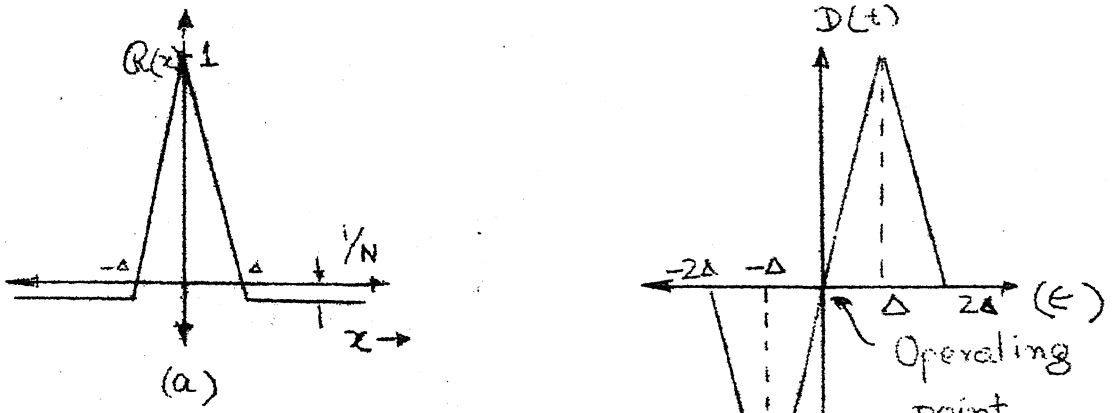


Figure 4.5 Binary Delay Lock Loop Using Shift Register Sequences

as shown in Fig. 4.6(a). The peaks of the correlation function will be delayed by the delay of the local reference signal. The difference of these two gives the DLL characteristic as shown in Fig. 4.6(b). Now, the summed correlator outputs are filtered and used to control the receiver's clock. The receiver's code will track the incoming code at a point half way between the maximum and minimum of the composite correlator output. So the incoming signal will locked on to the (n-1)th stage and will keep on tracking the incoming signal.

The block schematic of the implemented system is shown in Fig. 4.7.

The received data stream is EXORed with the output of the 7th and the 5th stage of the maximal length PN sequence generator. The output of the 7th stage correlator is inverted and added with the other correlator output. This is the same thing as to take difference of the two correlator outputs. The only difference is, it is biased on a d.c. value. This output is fed to a simple RC low pass filter. Next block controls the loop gain. The VCO(MC 4024) operates linearly between 2.5V to 5V. So the output of the loop gain and operating point control unit is adjusted so that the operating point is at the linear region of the VCO. The



4.6 (a) Correlation function

Figure 4.6. (b) DLL characteristic

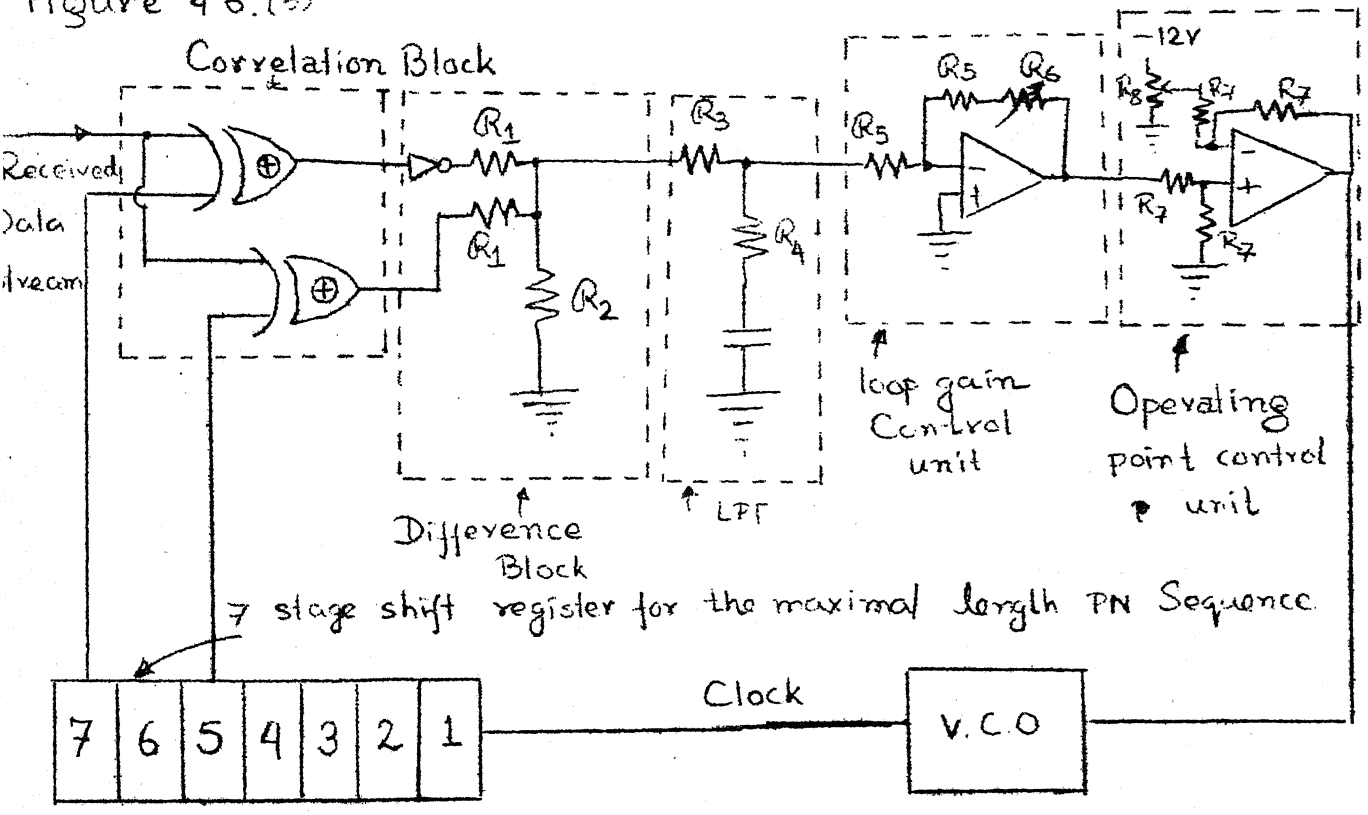


Figure 4.7. Functional Blocks of the implemented Delay Lock Tracking System

VCO output is fed to the clock of the shift register stages. According to the delay estimate the clock gets adjusted and the incoming PN sequence gets locked on to the 6th stage of the shift register and keeps on tracking the incoming sequence.

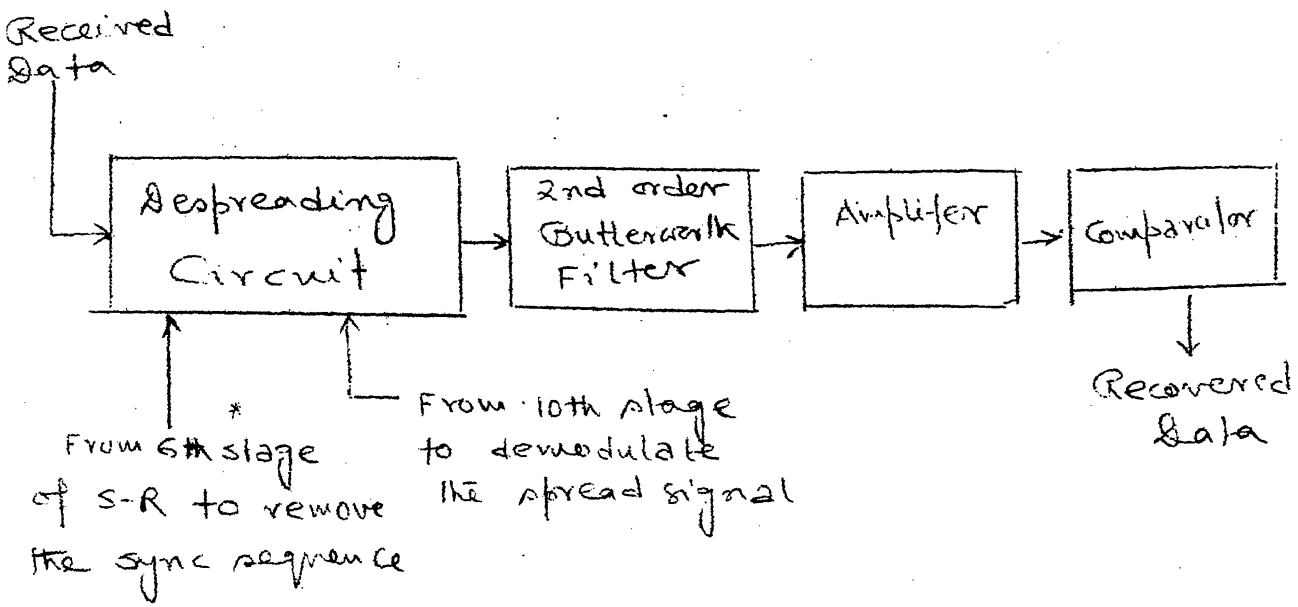
When the data is '1' the PN sequence gets complemented. So the slope of the DLL characteristic gets reversed. Thus the loop becomes unstable and goes out of lock. To get rid of this, a bit shifted PN sequence is sent with the modulated data. Thus the locking is achieved with the bit shifted PN sequence.

4.3.2 Data Recovery:

After despreading the received signal, it is integrated over the data period. Then properly putting the comparator threshold the data is recovered. Fig. 4.8 shows the block schematic of the implemented data recovery circuit.

The received signal is $PN' + PN \oplus D + N_1$ where PN' is the bit shifted PN sequence for synchronization. $PN \oplus D$ is the modulated data N_1 is the interference noise caused by the other users.

As DLL gets locked with PN', the PN' is removed first by EXORing the 6th stage output with the received signal.



* The sync signal gets locked with the (n-1)th stage ie 6th stage of the shift register. As 4 bit shifted signal was used for spreading the data, the despreading has to be done by the 10th stage output.

Fig 4.8. The Data Recovery Circuit

Now this signal,is despreaded by EXORing it with PN. This demodulated data and the interference noise is fed to the 2nd order Butterworth filter. The filter output is amplified to the required level. Putting the comparator (710) threshold properly data is recovered.

The details circuit diagram is given in Appendix A1 (Figure A1.3).

CHAPTER 5

PERFORMANCE EVALUATION

This chapter gives the estimate of bit error probability of the SSMA system as a function of number of users. The performance of the network is evaluated for finite number of users.

5.1 BIT ERROR PROBABILITY IN BASEBAND SSMA SYSTEMS:

It is assumed that the interference due to the other users are the only source of bit error. So neglecting the channel noise the equation (4.2) in Sec. 4.1 is rewritten as

$$Z_i = A \int_0^T b_i(t)dt + \sum_{\substack{k=1 \\ k\neq i}}^{K} A \int_0^T b_k(t-\tau_k)a_k(t-\tau_k)a_i(t)dt \qquad (5.1)$$

Now the signal to noise ratio is calculated as in [12] by finding the mean square value of the multiple access interference term. (See Appendix 2). The signal to noise ratio is given by

$$SNR_i = \left\{(3N^3)^{-1} \sum_{\substack{k=1 \\ k\neq i}}^{K} \gamma_{k,i}\right\}^{-1/2} \qquad (5.2)$$

where N is the length of the PN sequence and $\gamma_{k,i}$ is the average interference parameter.

The approximate probability of error (suggested by Pursley [12]) is $P_e \approx 1-\phi(SNR_i)$, ϕ is the Gaussian distribution function. The accurate evaluation of P_e can be found in [14] and in that the author shows this approximate probability of error is very close to the accurate value when number of users is large.

From the table of $\gamma_{k,i}$ in Appendix 2 [15] the SNR for different number of users is calculated. From this the bit error probability is evaluated. Table 5.1 shows signal to noise ratio and bit error probability with the number of users.

The results in Table 5.1 is for linear addition case i.e. all the transmitted signals are added to give a multi-level signal at the receiver input. But in our system all the transmitted signals are ORed. The composite signal at the receiver input is binary. The scenario is same as to put an amplitude limiter at the input of the receiver. So there will be a 2 db loss in the SNR [21]. Table 5.2 gives the bit error probability and signal to noise ratio for binary received signal with 2 db loss.

5.2 MEASURED BIT ERROR PROBABILITY:

The test was conducted to find the bit error probability in a SS system due to the presence of other users. The

Table 5.1: Table for SNR and Bit Error Probability with number of users

No. of users	SNR (db)	Bit Error Probability (P_e)
1	10.86	$< 10^{-10}$
2	9.57	$< 10^{-10}$
3	8.77	$< 10^{-10}$
4	8.21	0.2×10^{-10}
5	7.8	0.87×10^{-9}
6	7.4	0.19×10^{-7}
7	7.07	0.17×10^{-6}
8	6.8	0.79×10^{-6}
9	6.57	0.34×10^{-5}

Table 5.2: Table for SNR and Bit Error Probability with no. of users considering 2 dB loss

No. of users	SNR (db)	Bit Error Probability (P_e)
1	8.86	$< 10^{-10}$
2	7.57	0.6×10^{-8}
3	6.77	0.8×10^{-6}
4	6.21	0.13×10^{-4}
5	5.8	0.72×10^{-4}
6	5.4	0.23×10^{-3}
7	5.07	0.68×10^{-3}
8	4.8	0.13×10^{-2}
9	4.57	0.19×10^{-2}

experimental set up is shown in Fig. 5.1. The data rate is 4.8 K baud. A 7 bit PN sequence is send as data. User is simulated by sending some other 127 bit PN sequences with the SS modulated data. The users are added in a nonlinear manner by ORing.

For synchronous users the bit error probability is very low. It was found that in 15 minutes the number of bits in error is zero. No error was detected upto seven users.

For asynchronous users the bit error probability increases rapidly with the number of users. As the number of users are increased the lock range of the DLL gets decreased. The DLL tends to go out of lock on the addition of the 7th user. For asynchronous users the measured results are given in Table 5.3.

Table 5.3: Measured Bit Error Probability

No. of users	Bit Error Probability
1	$< 10^{-10}$
2	0.1×10^{-7}
3	0.65×10^{-5}
4	0.74×10^{-4}
5	0.31×10^{-3}
6	0.11×10^{-2}

5.3 THE THROUGHPUT AND DELAY CHARACTERISTICS:

The bit and packet error rates with m simultaneous transmissions are defined as $P_e(m)$ and $P_\varepsilon(m)$ respectively. It is assumed that all transmissions are received with equal power and multiuser interference is the only cause of errors. So the correct packet probability $P_c(m) = 1 - P_\varepsilon(m)$. SSMA codes are sufficiently pseudorandom to permit the assumption that errors in different packets occur independently. The steady state probability distribution of the composite arrivals be $f(m)$, so the steady-state throughput β is the expected number of successful transmissions per unit time. Thus

$$\beta = \sum_{m=1}^{\alpha} m\, P_c(m)\, f(m)$$

Now for finite number of users the composite arrival distribution is Binormal so

$$f(m) = \binom{N}{m} p^m (1-p)^{N-m} \text{ for } m \leq N$$

when N = number of users

p = the probability of transmission of a new packet

Let y = offered traffic i.e. average number of attempted transmissions at a time

$\therefore$ p = y/N

So

$$\beta = \sum_{m=1}^{N} m \binom{N}{m} \left[\frac{y}{N}\right]^m \left[1 - \frac{y}{N}\right]^{N-m} P_c(m) \qquad (5.10)$$

where $P_c(m) = [1-P_e(m)]^L$, L is the length of the packet. Now for our case L = 2096, and N = 9. The correct packet probabilities for the calculated and measured cases are given in Table 5.4. From equation (5.10) the throughputs in packets/unit time for different offered traffics are calculated. The Fig. 5.2 shows the offered traffic vs. the throughput. It is found that the measured throughput is quite comparable to the calculated results with 2 db loss. It is seen that arrival of more than four packets/unit time makes the system unstable i.e. any further increase in offered traffic decreases the throughput.

Another important performance parameter is the average delay (D). A measure of this can be obtained in terms of retransmissions per successful packet.

$$r_{av} = \frac{y-\beta}{\beta} = \frac{y}{\beta} - 1$$

So the average delay D = r_{av} X average rescheduling delay. Fig. 5.3 shows the throughput (β) in packets/unit time vs. average retransmission delay (r_{av}). As the throughput goes beyond its maximum value the delay tends to increase very rapidly.

Table 5.4: Correct Packet Transfer Probability

No. of users(m)	$P_c(m)$ considering 2 db loss	$P_c(m)$ of the system
1	1	1
2	1	1
3	0.998	0.986
4	0.973	0.856
5	0.859	0.522
6	0.617	0.099
7	0.240	0
8	0.065	0
9	0.018	0

(Probabilities > .999 set equal to 1)

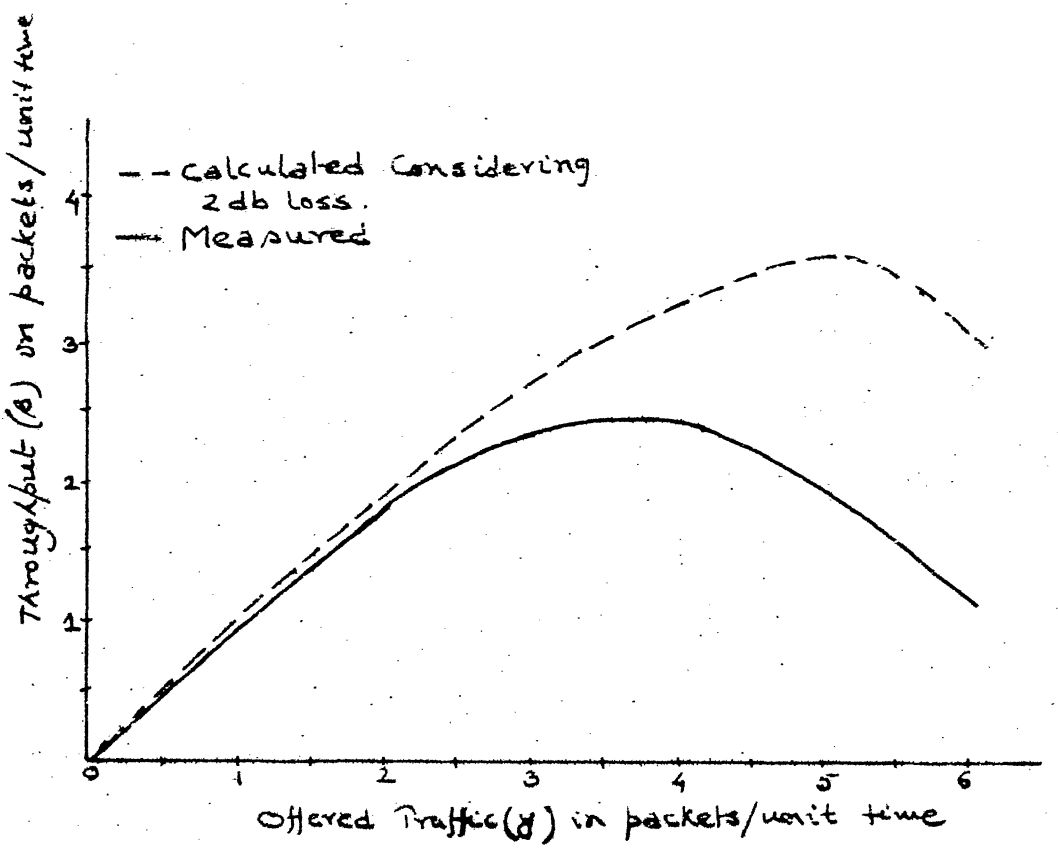


Fig 5.2 Offered Traffic Vs. Throughput

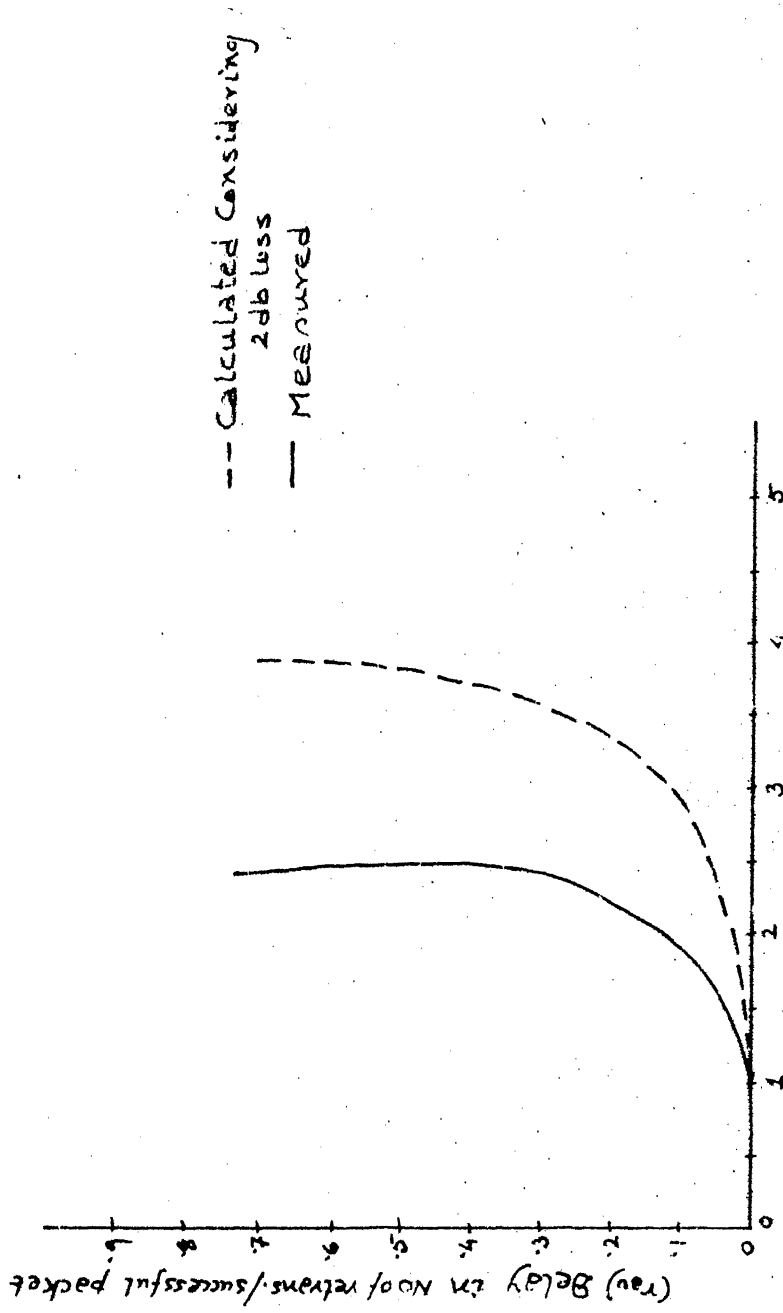


Fig 5.3 Throughput Vs. Average Retransmission Delay

CHAPTER 6

CONCLUSION

This concluding chapter compares the performance of SSMA system with other popular random access techniques. It also suggests the improvements that can be easily incorporated.

6.1 COMPARISON WITH OTHER TECHNIQUES:

The throughput and delay characteristic of the spread - spectrum network improves considerably over the slotted ALOHA and CSMA techniques. It is found from the Fig. 6.1, the maximum throughput of the system is 2.4 packets/unit time where the same for slotted ALOHA is 0.18 packets/slots and 0.56 packets/unit time for 1 persistance CSMA. The average delay is also quite low because more than one packet can be transferred at a given time. Fig. 6.2 shows the throughput vs. average number of retransmission for finite terminal. All these advantages are achieved at the cost of bandwidth. By increasing the processing gain i.e. the bandwidth of the spread signal a further improvement in performance can be achieved. The bandwidth is not a constraint in local area networks. The extra hardware requirement is the only overhead, specially the receiver circuit involves a lot of analog

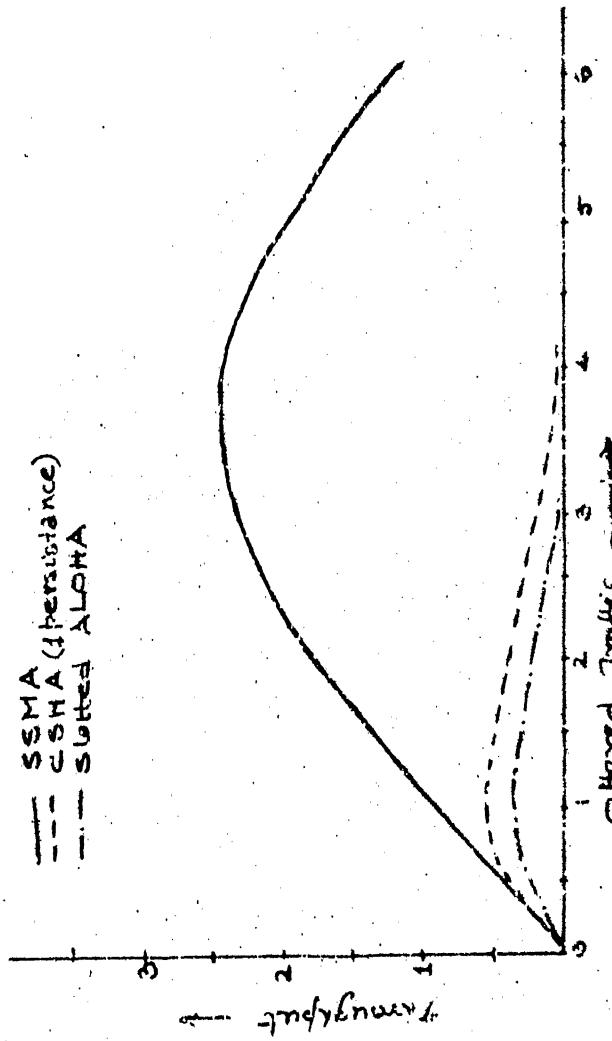


Fig 6.1 Throughput in Different Techniques.

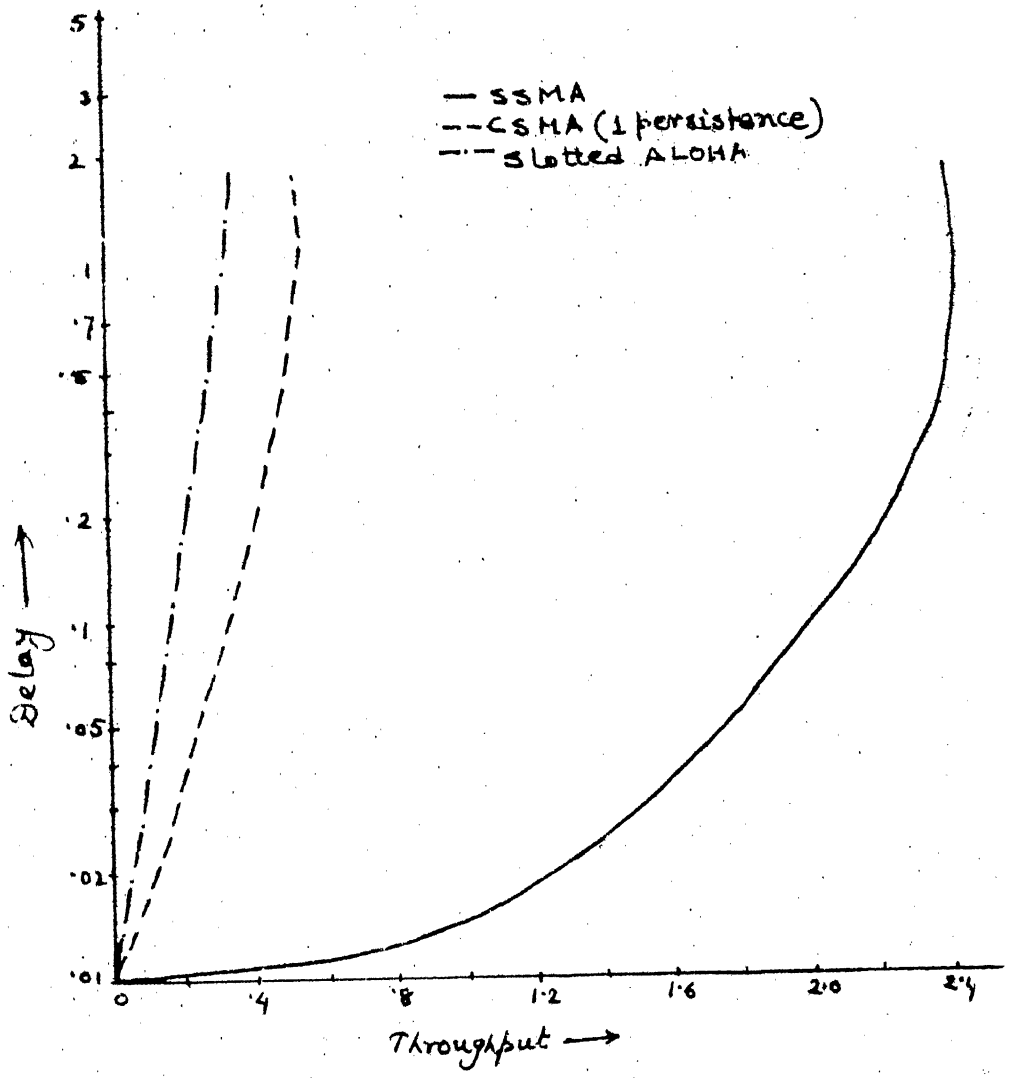


Fig 6.2 Delay in Different Techniques

processing. To introduce a new user to the network, its transmitter has to be provided to all other stations. The advantages of the system can be really appreciated if the number of users are very large and the traffic is bursty in nature. As spread spectrum systems are inherently immuned to noise, the scheme can be efficiently implemented in error prone channels.

6.2 SUGGESTIONS FOR FURTHER IMPROVEMENTS:

i) The data-link layer protocol has been implemented in such a manner that the implemented stop and wait protocol can be modified to go back n protocol. The control byte of I-frame has got three unused bits. These can be used to incorporate piggy-backing feature.

ii) The error correcting codes can be incorporated, which will reduce the bit error probability. In view of the feasibility of recovery of packets the throughput can be further increased.

iii) The packet length may be optimized depending on the channel characteristic.

iv) The baud rate can be increased to 64K baud. By using the inbuilt clock recovery circuit of the 8273 the maximum baud rate achieved is 9.6K baud. So an attempt was made to extract the clock from the recovered data. But as interrupt driven data transfer has been used the baud rate cannot go beyond 10 K baud. So using DMA and the implemented clock recovery circuit given in Appendix 3 the baud rate can be increased.

v) The upper layers of the ISO-OSI model have not been touched. These layers can be elaborated.

vi) For simplicity, the correlators used in receiver are digital. An analog multiplier will improve the DLL performance.

vii) The SSM system will perform better if two seperate maximal length PN sequences are used for synchronization and modulation. The transmitter will have two different PN sequence generators. The receiver circuit has to be modified to achieve these two sequences in correct phase and frequency. Referring to Fig. 6.3 the sync. sequence has to be recovered using DLL. A decoder (same as in transmitter) with PLL can be used to get the modulating PN sequence. The despreading and data recovery circuit is same.

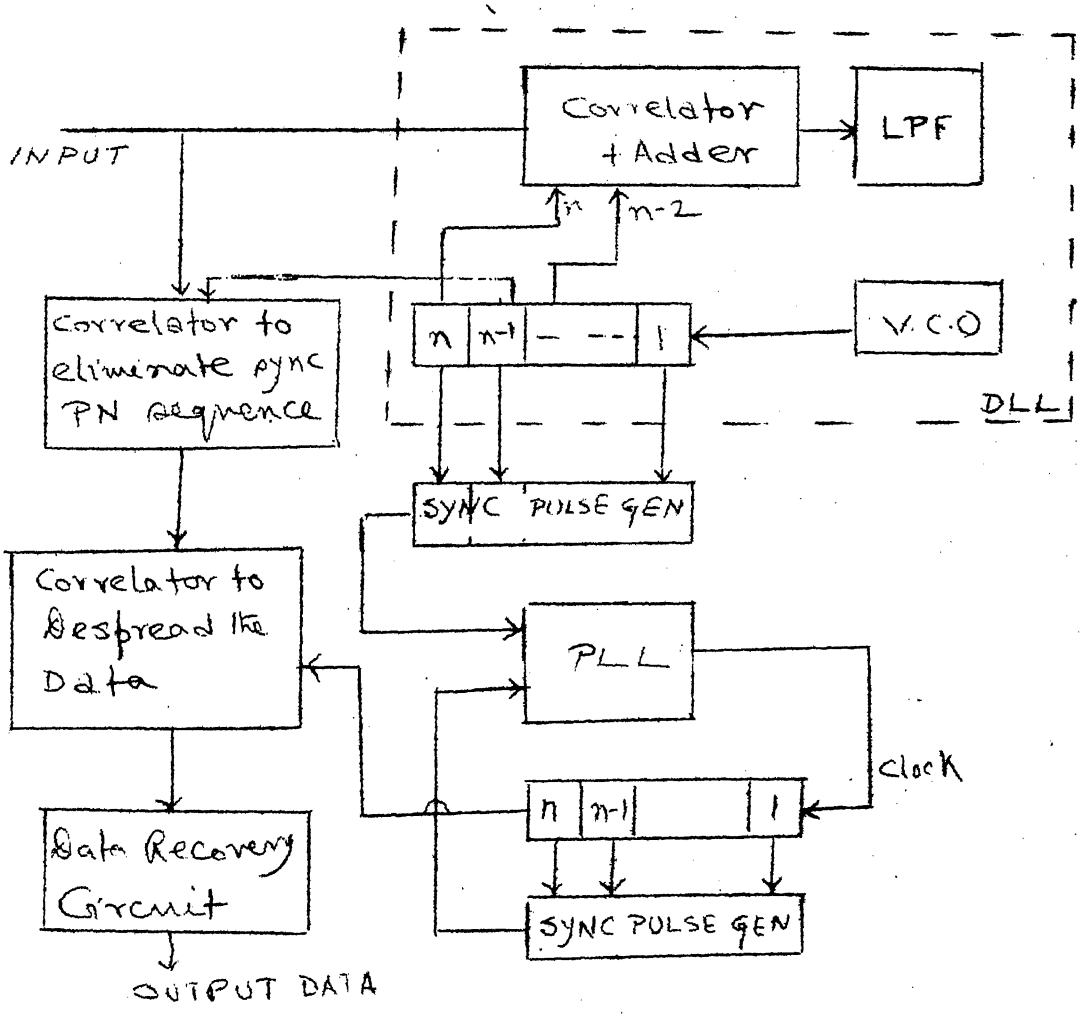


Fig 6.3 SS Receiver

6.3 SSMA IN OTHER NETWORK TOPOLOGIES:

The SS system can also be used in broadcast type local area networks. It can also be implemented with data loop to achieve the multiple access capability.

6.3.1 Broadcast Like Network:

The network will have a controller station (primary) and other stations (secondaries) will be connected to it. Referring to Fig. 6.4, when any secondary wants to communicate with another secondary the primary will switch the packets. When primary transmits, all the secondaries listen to it and addressee receives it. As there is no contention, the primary transmitted signals need not be SS modulated. So all the secondaries will have only transmitter. The secondaries will modulate the data with its own code. The primary needs to have a dual processor architecture. The spread received signals has to be despreaded and loaded in memory. The front end processor will transfer data from memory to the main processor. The main processor will support the protocols for error free transmission and switching.

6.3.2 Data Loops:

Referring to Fig. 6.5, SSMA data loop will have a sync. node (primary) and other secondary nodes. The sync. node

transmits a PN sequence as sync. sequence over the loop to establish synchronization. Each node will be assigned a code which is bit shifted sequence of the sync. sequence. The transmitter clock is extracted from the sync. signal. A transmitting node modulates the data with a code assigned to receiving node and sends with the sync. signal. The composite signal circulates in the loop. The receiver locks on to the sync. signal and despreads the data using proper bit-shift information. The advantages of a SS multiple access loop are:

i) Any station can communicate with any other station without going through primary.

ii) More than one users can access the loop simultaneously.

iii) All the nodes need a single sync. signal where the implemented system needs one sync. signal for each node.

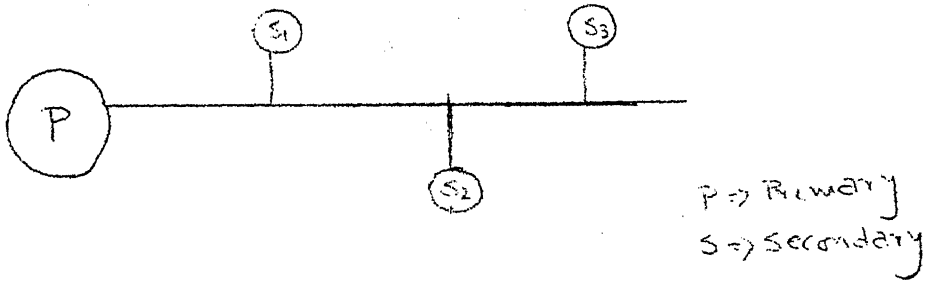


Fig 6.4 Broadcast Like Topology

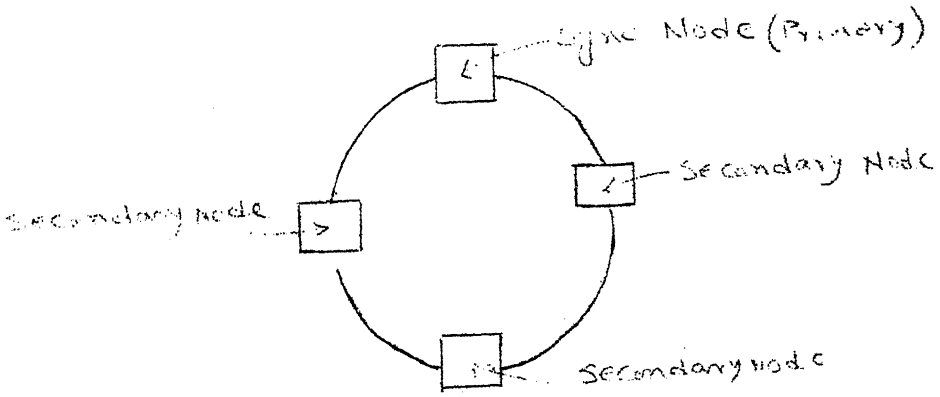
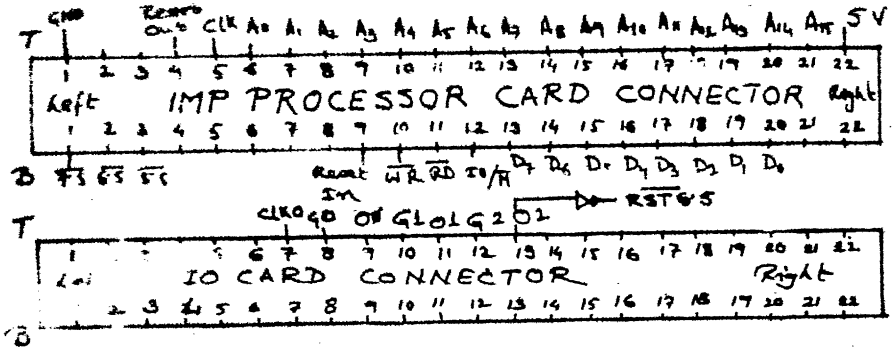
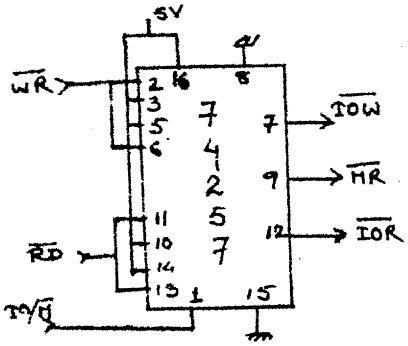
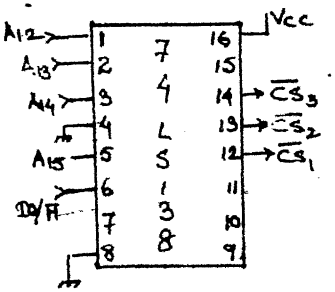
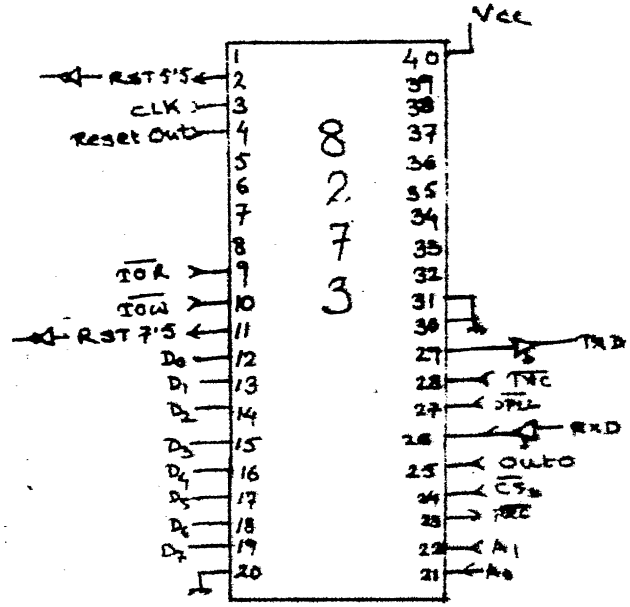


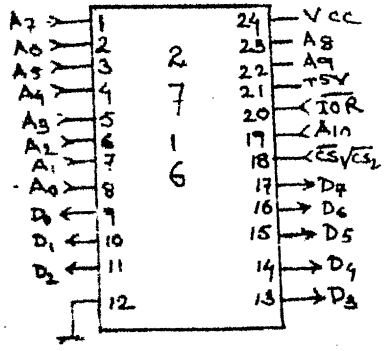
Fig 6.5 Loop Configuration





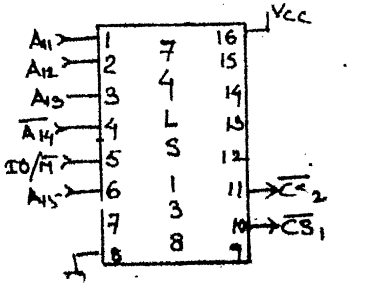


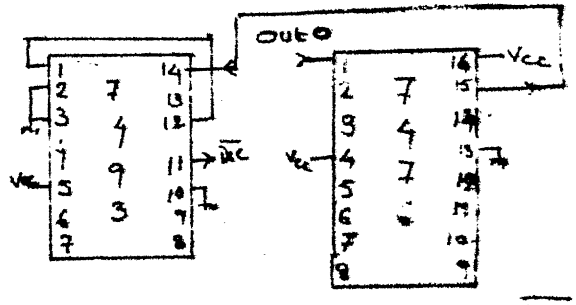




$\overline{CS}_1$ for E000H to E7FFH

$\overline{CS}_2$ for E800H to EFFFH





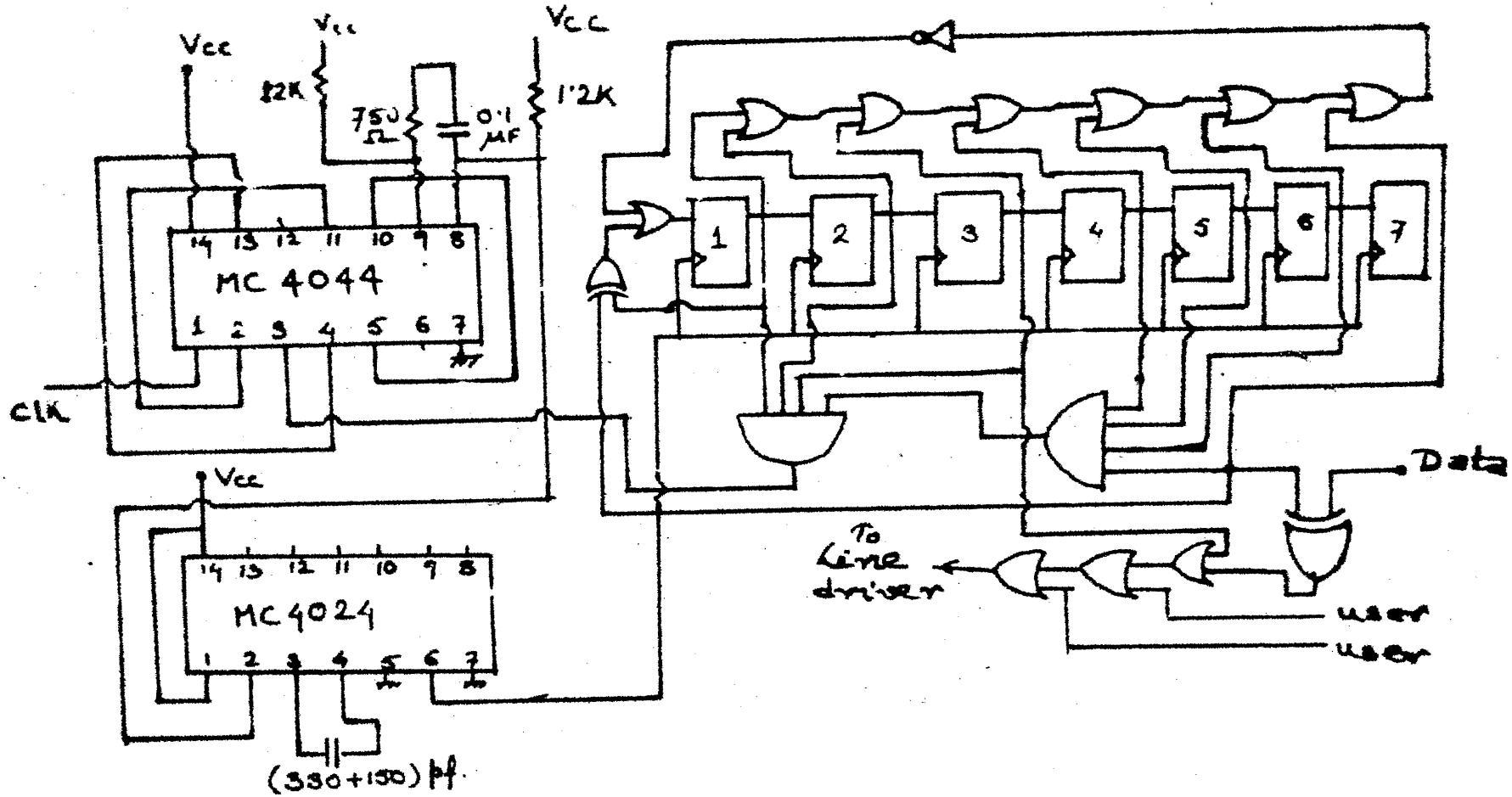


Fig A1.2 SSM Transmitter

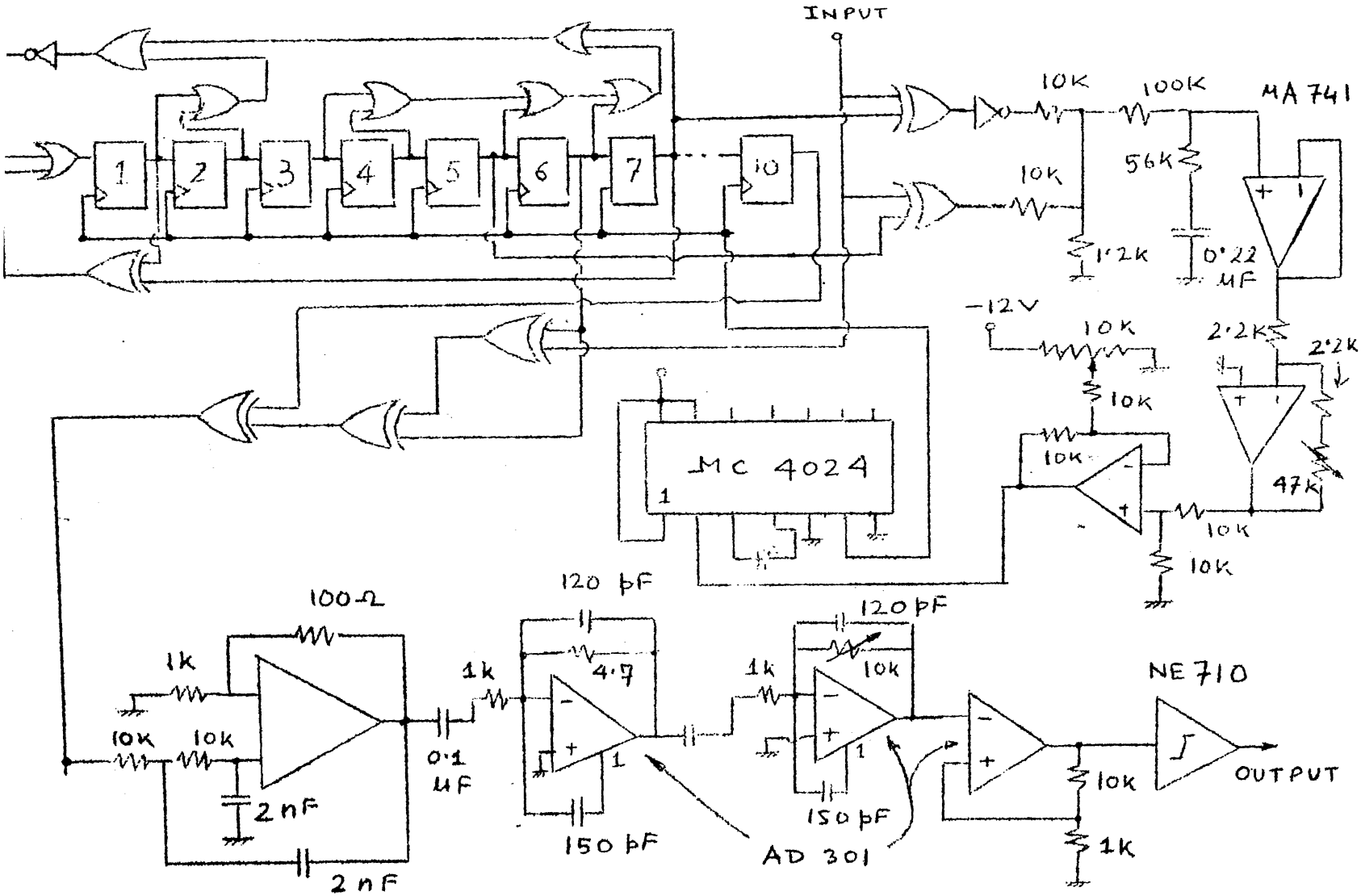


FIG SSM Receiver circuit diagram

APPENDIX 2

CALCULATION OF SNR IN SSM SYSTEMS

The equation (5.2) in Sec. 5.1 is rewritten as

$$Z_i = Ab_{i,o}\,T + \sum_{\substack{k=1 \\ k\neq i}}^{K} \left(b_{k,-1}\, i \int_0^{\tau_k} a_k(t-\tau_k)a_i(t)dt \right.$$

$$\left. + b_{k,o} \int_{\tau_k}^{T} a_k(t-\tau_k)a_i(t)dt \right) \qquad \text{(A2.1)}$$

This is shown in Fig. A2.1

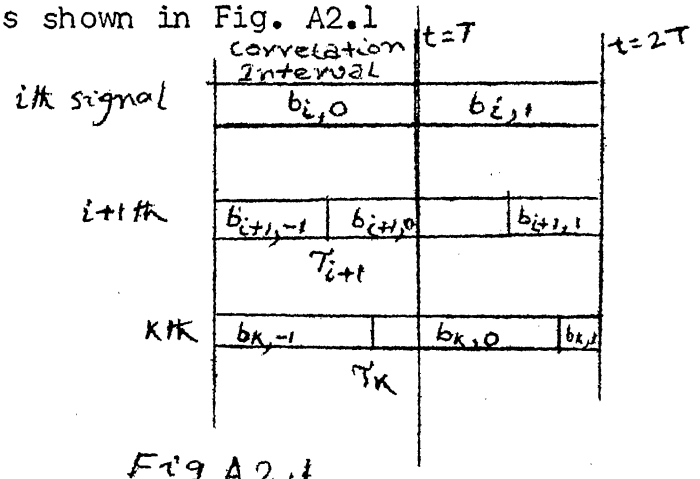


Fig A2.1

The integrals in eqn. (A2.1) are given by

$$R_{k,i}(\tau) = \int_0^{\tau_k} a_k(t-\tau_k)a_i(t)\,dt$$

$$R_{k,i}(\tau) = \int_{\tau_k}^{T} a_k(t-\tau_k)a_i(t)dt$$

These are the partial cross-correlation functions. Their effect on the receiver output is via the quantity

$$I_{k,i}(b_k, \tau_k) = T^{-1}[b_{k,-1}R_{k,i}(\tau) + b_{k,o}\hat{R}_{k,o}(\tau)] \tag{A2.2}$$

which is the multiple-access interference at the output of the ith receiver due to the kth signal. So,

$$Z_i = A[b_{i,o} + \sum_{i=1}^{K} I_{k,i}(b_k, \tau_k)] \tag{A2.3}$$

We can assume b, τ to be independent as they arise from independent physical phenomena. Now

$$E[I_{k,j}(b_k, \tau_k)] = 0 \text{ since } P(b_{k,-1}=1) = P(b_{k,o}=1)=1/2$$

The variance of $I_{k,i}(b_k, \tau_k)$ is given by

$$\sigma_{k,i}^2 = \text{Var}[I_{k,i}(b_k, \tau_k)]$$

$$= T^{-2} \int_0^T T^{-1} [R_{k,i}^{\nu}(\tau) + \hat{R}_{k,i}^{\nu}(\tau)]d\tau \tag{A2.4}$$

Defining the quantities

$$m_{k,i} = \int_0^T R_{k,i}^{\nu}(\tau)d\tau, \quad \hat{m}_{k,i} = \int_0^{\tau} R_{k,i}^{\nu}(\tau)d\tau$$

we have

$$\sigma^2_{k,i} = T^{-3} \left(m_{k,i} + \hat{m}_{k,i}\right) \qquad \text{(A2.5)}$$

Using the expressions given in [9], $\sigma^2_{k,i}$ can be written as

$$\sigma^2_{k,i} = (3N^3)^{-1} \left[2\mu_{k,i}(o) + \mu_{k,i}(1)\right] \qquad \text{(A2.6)}$$

where $\mu_{i,j}(n)$ is given by

$$\mu_{i,j}(n) = \sum_{i=1-N}^{N-1} C_i(\ell)\, C_j(\ell+n)$$

where $C_i(\ell)$, $C_j(\ell)$ are the aperiodic autocorrelation functions of the ith and jth signature sequence.

$C_{k,i}(\cdot)$ is defined by

$$C_{k,i}(L) = \begin{cases} \sum_{j=o}^{N-1-L} a_j(k) \cdot a^{(i)}_{j+L}, & 0 \le L, \le N-1 \\ \sum_{j=o}^{N-1+L} a^{(k)}_{j-L}\, a_j^{(i)}, & 1-N \le L < 0 \end{cases} \qquad \text{(A2.7)}$$

From (A2.6)

$$\sigma^2_{k,i} = (3N^3)^{-1} \sum_{k=1}^{K} r_{k,i}$$

$r_{k,i}$ is the average interference parameter.

$$\text{So } (SNR)_i = \left[(3N^3)^{-1} \sum_{k=1}^{K} r_{k,i}\right]^{-1/2}$$

Table A2.1: Average interference parameters ($r_{k,i}$)

Tap	211	217	235	247	277	357	323	203	325
211	41214	33622	32722	32022	29070	34394	32978	31486	30250
217		40222	30954	31406	31446	29554	32666	34118	32066
235			42046	32570	33714	32974	32246	33106	33310
247				40326	34054	32002	29546	30486	33634
277					41294	31146	31938	33934	34202
357						41958	31258	30806	30770
323							39870	33538	33598
203								39902	32442
325									42894

The tappings are in octal.

APPENDIX 3

THE CLOCK RECOVERY CIRCUIT

A delay and multiply synchronizer circuit has been made to extract clock from the received data pattern. The block schematic of the circuit is shown in Fig. A3.1.

A delay of half the clock is obtained by two monostables and an inverter. Narrow-band band-pass filter is realized by a PLL. Its pull-in range and lock ranges are critically adjusted for a particular data rate.

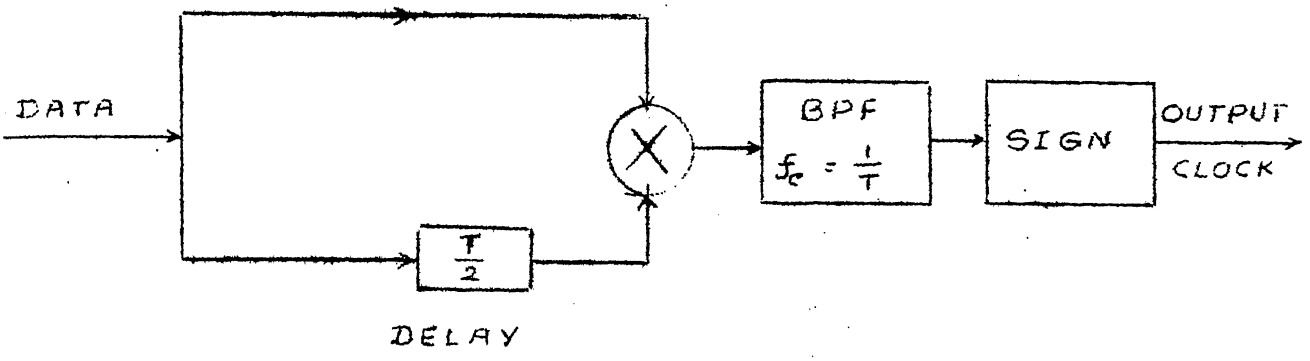


FIG A 3·1 SCHEMATIC DIAGRAM

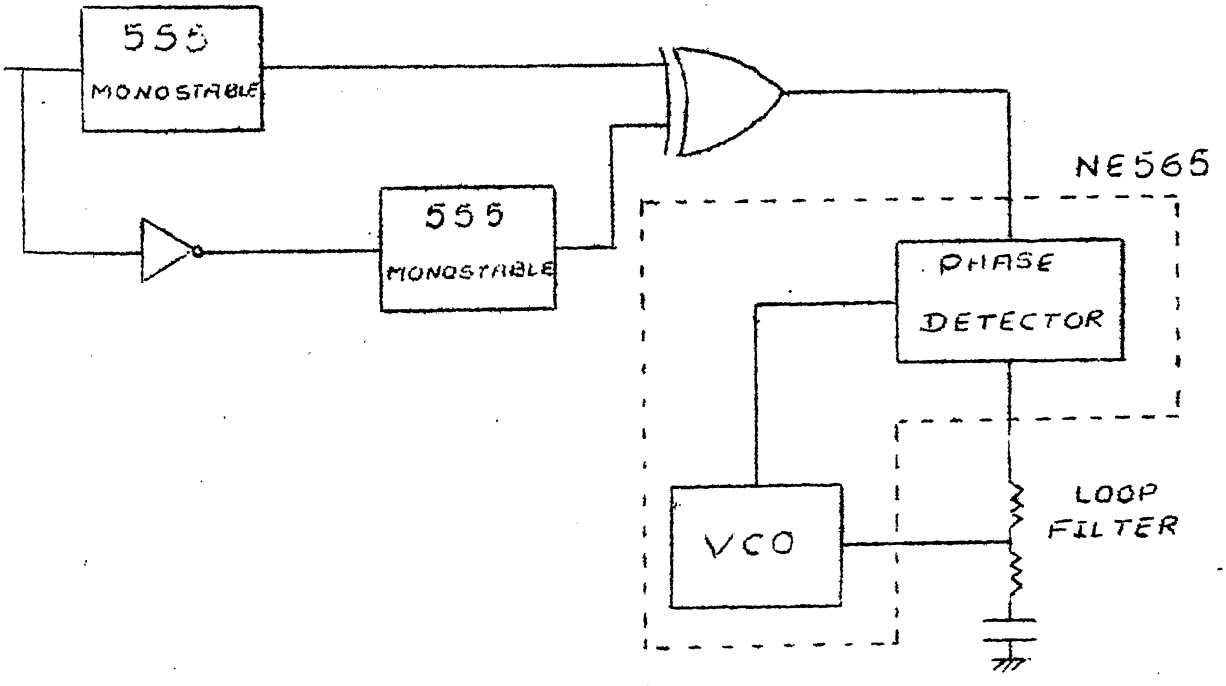


FIG A 3·2

IMPLEMENTED CLOCK RECOVERY CKT

REFERENCES

[1] M. Schwartz, Computer-Communication Network Design and Analysis. Englewood Cliffs, N.J. Prentice Hall, 1977.

[2] A.S. Tanenbaum, Computer Networks, Englewood Cliffs, N.J. Prentice-Hall, 1981.

[3] N. Abramsen, 'The throughput of Packet Broadcasting Channels', IEEE Trans. Commun., vol. COM-25, pp. 117-128, Jan. 1977.

[4] L. Kleinrock and S.S. Lam, 'Packet Switching in a Multi-access Broadcast Channel: Performance Evaluation', IEEE Trans. Commun, vol. COM-23, pp. 410-423, April,1975.

[5] D.W. Davies, D.L.A. Barber, W.L. Price and C.M. Solomo-ridies, Computer Networks and Their Protocols, N.Y. John Willey, 1979.

[6] R.C. Dixon, Spread Spectrum Systems, N.Y. John Willey, 1976.

[7] R.A. Scholtz, 'The Spread Spectrum Concept', IEEE Trans. Commun, vol. COM-25, pp. 748-755, August,1977.

[8] R.L. Pc Pickholtz, D.L. Schilling, L.B. Milstein, 'Theory of Spread-Spectrum Communications - A tutorial', IEEE Trans. Commun., vol. COM-30, pp. 855-884, May,1982.

[9] G. Longo, Ed., Multi-User Communication Systems, N.Y. Springer-Verlag.

[10] D. Raychaudhuri, 'Performance Analysis of Random Access Packet-switched Code Division Multiple Access Systems', IEEE Trans. Commun., vol. COM-29, pp. 895-901, June 1981.

[11] A.R.K. Sastry, 'Effect of Acknowledgement on the Performance of Slotted ALOHA Code Division Multiple Access Systems', Presented at the IEEE MILCOM, Oct. 31-Nov. 2,1983, Washington D.C.

[12] M.B. Pursley, 'Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication - Part I: System Analysis', IEEE Trans. Commun, vol. COM-25, pp. 795-799, August 1977.

[13] M.B. Pursley and D.V. Sarwate, 'Performance Evaluation for Phase-coded Spread-spectrum Multiple-access Communication - Part II: Code Sequence Analysis', IEEE Trans. Commun, vol. COM-25, pp. 800-803, August 1977.

[14] K. Yao, 'Error Probability of Asynchronous Spread Spectrum Multiple Access Communication System', IEEE Trans. Commun. vol. COM-25, pp. 803-809, August,1977.

[15] M.B. Pursley and H.F.A. Roefs , 'Numerical Evaluation of Correlation Parameters for Optimal Phases of Binary Shift-register Sequences', IEEE Trans. Commun. vol. COM-27, pp. 1597-1604, October 1979.

[16] H. Zimmermann, 'The OSI Model of Architecture for Open Systems Interconnection', IEEE Trans Commun., vol. COM-28, pp. 425-432, April,1980.

[17] R.M. Metcalfe and D.R. Boggs, 'Ethernet: Distributed Packet Switching for Local Computer Networks', Commun. ACM, vol. 19, pp. 395-404, July,1976.

[18] F.A. Tobagi, 'Multi-access Protocols in Packet Communication Systems', IEEE Trans. Commun. vol. COM-28, pp. 468-488, April,1980.

[19] R.E. Khan, S.A. Gronemeyer, J. Burchfill and R.C. Krinzelman, 'Advances in Packet Radio Technology', Proc. IEEE, vol. 66, Nov. 1978.

[20] B. Hirosaki, S. Hasegawa and A. Sawi,'Spread-spectrum Multiple Access Data Loop', pp. G8.5.1 - G8.5.6, 1981.

[21] Spilker, Digital Communication with satellite, N.Y. McGraw Hill, 1968.

[22] A. Osborne, An Introduction to Microcomputers, vol. 3, Some Real Support Devices, California, Osborne and Associates Inc. 1978.

[23] INTEL, Microprocessor and Peripherals Handbook, 1983.

EE-1984-M-NAN-STU